UNIVERSIDAD CARLOS III DE MADRID

Departamento de Teoría de la Señal y Comunicaciones

DOCTORAL THESIS

BAYESIAN NONPARAMETRICS FOR CROWDSOURCING

Author: PABLO GARCÍA MORENO

Supervised by: FERNANDO PÉREZ CRUZ

ANTONIO ARTÉS RODRÍGUEZ

November 2015

Tesis Doctoral: BAYESIAN NONPARAMETRICS FOR CROWDSOURCING

Autor: Pablo García Moreno

Director: D. Fernando Pérez Cruz
D. Antonio Artés Rodríguez

Fecha: 13 de Noviembre de 2015

Tribunal

Presidente:

Vocal:

Secretario:

"The organic fundamental error of humanism
was that it desired to educate
the common people (on whom it looked down)
from its lofty stance instead of trying
to understand them and to learn from them."

Stefan Zweig
(*Erasmus the Rotterdam*)

Agradecimientos

"A mis padres"

Siendo honesto, el éxito de esta tesis no se debe tanto a mi esfuerzo y dedicación durante estos años, como a todas aquellas personas que en los momentos bajos (y hubo unos cuantos) no me dejaron tirar la toalla. Espero que a estas alturas sepan lo agradecido que les estoy, aunque como dicta el protocolo, intentaré a continuación plasmarlo por escrito, aunque sólo para poner en evidencia la terrible capacidad de expresar emociones de un Vallisoletano de a pie.

Empezaré por mis padres, que una mezcla de optimismo y pesimismo, y siempre velando por mí, han hecho posible esta tesis. Os lo debo todo. Seguiré por mi hermano pequeño, esa copia mejorada por la evolución de uno mismo que siempre me apoyo, pese a representar injustamente la oveja negra de la familia durante su infancia. Especial mención merece mi tía esa persona dotada de inteligencia y cabezonería más allá de los límites de la comprensión humana, y que desde que tengo uso de razón, ha hecho incansable uso de ello para impedir que me rindiese. Y como no abuelos, tíos, primos y demás familia que siempre han estado ahí, mostrando su apoyo y preocupación, preguntando sobre mis progresos repetidamente y sufriendo el calvario de escuchar largas y aburridas respuestas en las que trataba de explicarles de que iba todo este tinglado. Gracias a ellos esta tesis es un poco más comprensible, o eso espero.

Por supuesto estaré eternamente agradecido a todos mis amigos, y especialmente a aquellos que compartieron una buhardilla hoy día al borde de la desaparición y que siempre se preocuparon, sino por esta tesis, por todo lo demás.

Pero si alguien ha hecho posible el día a día de esta tesis, ha sido un grupo de personas excepcionales, tanto en lo profesional como en lo personal y quienes en muchos casos, pasaron del status de compañeros de trabajo a amigos en un tiempo récord.

Quedan por supuesto mi director de Tesis Antonio y mi supervisor Fernando, quienes han sido los que realmente me han proporcionado la oportunidad de re-

alizar esta tesis aguantándome (tarea no siempre fácil) todos estos años, e incluso solucionando mi futuro más allá de esta tesis con no merecidos elogios en forma de cartas de recomendación.

Otra de las personas que ha hecho posible este documento es Yee Whye Teh, a quien no tengo nada que decir excepto que nunca pensé que llegara a trabajar con alguien como él (obra de nuevo de no merecidas recomendaciones por personas a las que igualmente admiro)

Llegados a este punto, estoy seguro que dejo muchas personas por agradecer, así que me cubriré las espaldas copiando (aunque eche por la borda mi futuro como político) de la tesis de mi tía la siguiente frase a modo de chivo expiatorio: "a todos vosotros, que vosotros bien sabéis quien sois".

And of course, and against all odds given my terrible memory, I have not forgotten of Cynthia, the best thing that has happened to me in my PhD and that has been always encouraging me, supporting me, suffering me and of course, waiting for me to finally write these last words. This full stop is yours: .

This work was partially supported by the “Formación de Profesorado Universitario” fellowship from the Spanish Ministry of Education (FPU AP2009-1513).

Abstract

Supervised machine learning relies on a labeled training set, whose size is closely related to the achievable performance of any learning algorithm. Thanks to the progresses in ubiquitous computing, networks, and data acquisition and storage technologies, the availability of data is no longer a problem. Nowadays, we can easily gather massive unlabeled datasets in a short period of time. Traditionally, the labeling was performed by a small set of experts so as to control the quality and the consistency of the annotations. When dealing with large datasets this approach is no longer feasible and the labeling process becomes the bottleneck.

Crowdsourcing has been proven to be an effective and efficient tool to annotate large datasets. By distributing the labeling process across a potentially unlimited pool of annotators, it allows building large labeled datasets in a short period of time at a low cost. However, this comes at the expenses of a variable quality of the annotations, i.e. we need to deal with a large set of annotators of possibly unknown and variable expertise. In this new setting, methods to combine the annotations to produce reliable estimates of the ground truth are necessary.

In this thesis, we tackle the problem of aggregating the information coming from a set of different annotators in a multi-class classification setting. We assume that no information about the expertise of the annotators or the ground truth of the instances is available. In particular, we focus on the potential advantages of using Bayesian Nonparametric models to build interpretable solutions for crowdsourcing applications.

Bayesian Nonparametric models are Bayesian models which set a prior probability on an infinite-dimensional parameter space. After seeing a finite training sample, the posterior probability ends up using a finite number of parameters. Therefore, the complexity of the model depends on the training set and we can infer it from the data, avoiding the use of expensive model selection algorithms.

We focus our efforts on two specific problems. Firstly, we claim that considering the existence of clusters of annotators in this aggregation step can improve the

overall performance of the system. This is especially important in early stages of crowdsourcing implementations, when the number of annotations is low. At this stage there is not enough information to accurately estimate the bias introduced by each annotator separately, so we have to resort to models that consider the statistical links among them. In addition, finding these clusters is interesting in itself, as knowing the behavior of the pool of annotators allows implementing efficient active learning strategies. Based on this, we propose in two new fully unsupervised models based on a Chinese Restaurant Process prior and a hierarchical structure that allows inferring these groups jointly with the ground truth and the properties of the annotators.

The second problem is modeling inconsistent annotators. The performance of the annotators can be in-homogeneous across the instance space due to several factors like his past experience with similar cases. To capture this behavior, we proposed an algorithm that uses a Dirichlet Process Mixture model to divide the instance space in different areas across which the annotators are consistent. The algorithm allows us to infer the characteristics of each annotator in each of the identified areas, the ground truth of the training set, as well as building a classifier for test examples. In addition, it offers an interpretable solution allowing to better understanding the decision process undertaken by the annotators, and implement schemes to improve the overall performance of the system.

We propose efficient approximate inference algorithms based on Markov Chain Monte Carlo sampling and variational inference, using auxiliary variables to deal with non-conjugancies when needed. Finally, we perform experiments, both on synthetic and real databases, to show the advantages of our models over state-of-the-art algorithms.

Resumen

Todo aprendizaje máquina supervisado descansa sobre un conjunto de entrenamiento etiquetado cuyo tamaño muestral está directamente relacionado con el rendimiento final del algoritmo. Gracias a los avances en computación ubicua, redes y tecnologías de adquisición y almacenamiento de datos, la disponibilidad de datos con que entrenar estos algoritmos ha dejado de ser un problema. Actualmente, podemos facilmente reunir enormes conjuntos de datos no etiquetados en cortos periodos de tiempo. Tradicionalmente, el etiquetado de estos datos, era realizado por un pequeño conjunto de expertos a fin de controlar la calidad final y la consistencia de las anotaciones. Cuando nos enfrentamos a grandes conjuntos de datos, esta forma de proceder deja de ser factible, convirtiéndose el etiquetado en un cuello de botella.

Crowdsourcing ha probado ser una herramienta efectiva y eficiente para anotar grandes conjuntos de datos en aprendizaje máquina. Mediante la distribución del proceso de etiquetado a un, potencialmente ilimitado, conjunto de anotadores, permite construir grandes conjuntos de datos etiquetados en un corto periodo de tiempo y a un bajo coste. Sin embargo, todo esto tiene como precio una pérdida sobre el control de la calidad de las anotaciones. Nos enfrentamos ahora a un gran conjunto de anotadores cuya experiencia es variable y desconocida. En este nuevo escenario, métodos de combinación de las anotaciones para dar lugar a estimaciones fiables de la etiqueta verdadera son necesarios.

En esta tesis, abordamos el problema de agregar la información procedente de diferentes anotadores en un problema de clasificación multi-clase. Asumimos que no existe información disponible acerca de la experiencia de los anotadores o la etiqueta verdadera de las muestras. En concreto, nos centramos en las ventajas potenciales de usar modelos bayesianos no paramétricos para construir soluciones interpretables para aplicaciones de crowdsourcing.

Los modelos bayesianos no paramétricos son modelos Bayesianos que definen una probabilidad a priori sobre un espacio de parámetros con infinitas dimensiones.

Tras observar una muestra de entrenamiento finita, la probabilidad a posteriori termina usando un número finito de parámetros. Por tanto, la complejidad del modelo depende del conjunto de entrenamiento usado que es inferida a partir de los datos, evitando el uso de costosos algoritmos para selección de modelos.

Nos centramos en dos problemas específicos. En primer lugar, defendemos que tener en cuenta la existencia de grupos de anotadores en la etapa de agregación, puede mejorar el rendimiento global del sistema. Esto es especialmente importante en fases tempranas de la implementación del sistema de crowdsourcing, cuando el número de anotaciones en bajo. En esta fase no hay suficiente información para estimar con precisión el sesgo introducido por cada anotador por separado, por lo que tenemos que recurrir a modelos que tengan en cuenta las dependencias estadísticas entre los distintos anotadores. Además, encontrar estos grupos de anotadores es un problema interesante por sí mismo, pues el conocer el comportamiento de nuestros anotadores nos permite implementar estrategias eficientes de aprendizaje activo. Basándonos en esta hipótesis, proponemos dos nuevos modelos no supervisados haciendo uso de un prior *Chinese Restaurant Process* y un estructura jerárquica que nos permite inferir los grupos de anotadores así como sus propiedades y las etiquetas verdaderas.

El segundo problema es el modelado de anotadores inconsistentes. El rendimiento de los anotadores puede ser no homogéneo en el espacio muestral debido a diferentes factores tales como sus experiencias pasadas con casos similares. Para capturar este comportamiento, proponemos un algoritmo que usa un modelo *Dirichlet Process Mixture* con el objetivo de dividir el espacio muestral en diferentes areas en las cuales los anotadores son consistentes. El algoritmo nos permite inferir las características de cada anotador en cada una de las áreas identificadas, las etiquetas verdaderas de nuestras muestras de entrenamiento, así como construir un clasificador para futuras muestras. Además, ofrece una solución interpretable permitiendo una mejor comprensión del proceso de decisión adoptado por los anotadores, así como implementar estrategias para mejorar el rendimiento global del sistema.

Proponemos algoritmos de inferencia aproximada eficientes basados en muestreo *Markov Chain Monte Carlo* e inferencia variacional, usando variables auxiliares para lidiar con modelos de observación no conjugados cuando así se requiera. Finalmente, realizamos experimentos con bases de datos sintéticas y reales a fin de mostrar las ventajas de nuestros modelos con respecto al estado del arte.

Contents

List of Acronyms

AMT	Amazon Mechanical Turk
API	Application Programming Interface
BCNHA	Bayesian Combination of Non-Homogeneous Annotators
BMM	Bayesian Mixture Model
BNP	Bayesian Nonparametric
cBCC	Clustering based Bayesian Combination of Classifiers
CF	Collaborative Filtering
CRP	Chinese Restaurant Process
CSP	Cold Start Problem
DP	Dirichlet Process
DPMM	Dirichlet Process Mixture Model
EM	Expectation Maximization
EP	Expectation Propagation
FM	Factor Model
FMM	Finite Mixture Model

GMM	Gaussian Mixture Model
GP	Gaussian Process
hcBCC	Hierarchical Clustering based Bayesian Combination of Classifiers
HDP	Hierarchical Dirichlet process
HIT	Human Intelligence Task
HMM	Hidden Markov model
iBCC	Independent Bayesian Combination of Classifiers
IBP	Indian Buffet Process
MAP	Maximum a Posteriori
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MH	Metropolis Hasting
ML	Maximum Likelihood
MM	Mixture Model
RJMCMC	Reversible Jump Markov Chain Monte Carlo
SB	Stick Breaking
SMC	Sequential Monte Carlo

1 Introduction

1.1 Background and Motivation

The cornerstone of every supervised learning algorithm is a set of labeled instances called the training set (Ulusoy and Bishop, 2005). Supervised learning algorithms assume that this training set comes from an underlying probability distribution, and aim at building a function that is able to predict the labels of future unseen instances coming from the same distribution, i.e. the test set. For example, in medical diagnosis we may have a training set composed of samples coming from two groups: patients suffering a particular disease and a healthy control group. The goal is to use it to train a supervised learning algorithm, that will subsequently be able to predict whether a new patient is affected by the disease.

While most algorithms acknowledge the presence of noise in the observed instances, they do not take into account the existence of noise in the labeling process.

In practice, the labeling process is ultimately performed by humans, and therefore, it is subject to error.

For example, the diagnosis of tuberculosis is typically based on direct visual inspection of a spit sample coming from the patient. It is therefore interesting to build a classifier using these images. However, the sensitivity of the annotators range from $20\% - 80\%$, Steingart et al. (2006), and therefore, the corresponding training set will contain several errors.

Human errors can have different causes which have been deeply studied in psychology (Davies et al., 2013). In the context of labeling datasets, an annotator may present a bias that he involuntarily acquired during his training. In other occasions, the information provided to the annotator may be not enough to make an accurate decision or the description of the task may not be well defined. Sometimes the annotation task is intrinsically ambiguous, or the annotator may present a sloppy behavior due to fatigue, stress or any other biological cause. In every case, this translate into the appearance of errors in the labels of the training set, which ultimately harm the overall performance of the algorithm. Notice that even when the labeler is not human, this source of errors is still present, e.g. errors in automatic medical diagnostic tests.

Several theoretical (Lachenbruch, 1966, 1979; Bi and Jeske, 2010) and empirical studies (Nettleton et al., 2010; McDonald et al., 2003) about the negative effects of noise in the labeling process can be found in the literature. In particular, Zhu and Wu (2004) claim that the effect of the noise in the labeling process is generally more harmful than the noise in the instances. The authors give two main reasons. First, while we only have one label per instance, we generally have several features per instance, and the noise introduced from one to a limited number of them may have a limited impact in the overall performance. Second, when we build classifiers, the different observed variables may have a different importance in the process, being the label the most important one. Therefore, there is a need of designing algorithms that explicitly model the noise in the labeling process.

Another important limitation of supervised learning algorithms is the associ-

ated cost of building the labeled training set. Traditionally, this was done by a small set of experts to guarantee the quality and the consistency of the labels. With the advent of the era of *big data* (Che et al., 2013), the sizes of the datasets have experienced an exponential growth. It has been estimated that the worldwide data was approximately 1 ZB (10^{21} bytes) and that it will increase to 40 ZB by the year 2020 (see Figure 1.1).

By increasing the size of the training set, we can substantially improve the generalization properties of supervised learning algorithms (Halevy et al., 2009). The problem now is that there is a bottleneck in labeling those data (according to Gantz and Reinsel (2012), less than 3% of the digital data is labeled), which cannot be longer done by a small set of experts. In the example of the diagnosis of tuberculosis, a specialized medical center may receive hundreds of cases everyday, and with the increasing popularity of digital networks that allow centers to share their datasets, expecting one physician to look at every image to have a high quality consistent opinion is not realistic. Therefore, we are forced to distribute the task of labeling the training set.

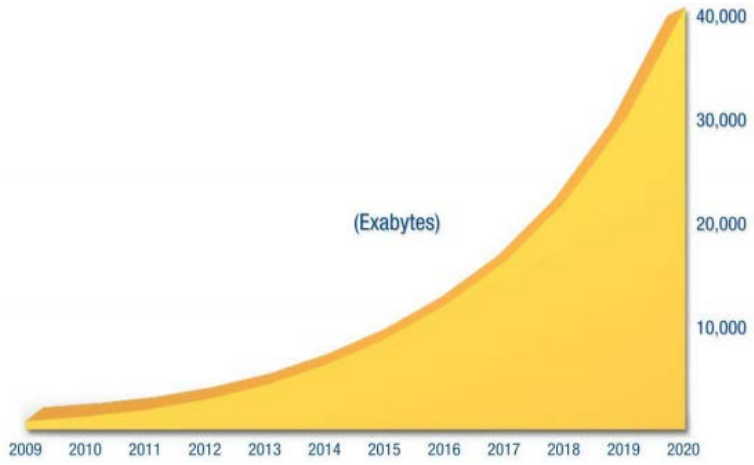


Figure 1.1: Estimation of the total amount of data in the digital universe. Borrowed from Gantz and Reinsel (2012)

By taking advantage of the ubiquity of the internet and cloud computing platforms, crowdsourcing has recently emerged as an effective way to efficiently distribute a task among a big pool of workers (Howe, 2006). In the context of machine

learning, crowdsourcing platforms have allowed to label in a distributed way big datasets in short periods of time at low cost.

When we outsource the labeling process to a crowd of annotators, for which we might not have sufficient information and whom might present variable performances, the quality of the labeling process can be put into question and the expected gains in the performance of our supervised learning algorithm might not realized by the increased labeled training data. In our example, instead of assigning the task to a hired physician for which we know his performance (we have access to his resume, we may have information about his performance in previous diagnostics, etc.), now we may upload the images to an online platform and allow anonymous physicians to label them.

Aggregation has emerged as a powerful technique to increase the overall quality of assigning a label. The advantage of crowdsourcing is that, even though the annotations provided by the physician may present a lower quality, we can have the opinion of a large pool of them at a low cost. If we combine the opinion of all physicians in a sensible way we can improve the overall performance of the systems, and obtain a high quality estimation of the ground truth. One of the earliest claims that theoretically support aggregation is the *Jury Theorem* (Condercet, 1785), which states that if we have a pool of annotator, each of them which a probability of choosing the correct answer between two options greater than 0.5, a majority voting strategy will provide the correct answer with probability 1 when the number of annotators tends to infinity. When we have a finite number of physicians with different performances, that have labeled different subsets of a set of instances related by an underlying distribution, majority voting stop guaranteeing correctness, and therefore, there is room for improvement.

In this thesis we focus on the aggregation problem in a multi-class classification scenario. Specifically, we assume that we observe a set of labels provided by different annotators for a training set. The goal is to provide an estimate of the ground truth, the properties of the annotators and optionally, a classifier for future instances. In the next section, we specifically state our contributions.

1.2 Contributions

In this thesis, we develop Bayesian Nonparametric (BNP) models to capture different aspects of crowdsourcing applications and in particular, to address the problem of how to combine the labels provided by a pool of annotators whose expertise or intentions are unknown. We summarize our contributions below.

1.2.1 Identifying Communities of Annotators

Latent variable models to combine the labels provided by a set of unreliable error-prone annotators have become an active research line. However, the high number of parameters in these models make them prone to overfitting, yielding poor results in practical scenarios where the information collected from the annotators may be scarce.

This is specially important in early stages of the deployment of crowdsourcing applications. In this stage, the number of annotators that have labeled a particular instance is very small. Likewise, the number of instances labeled by a particular annotator is also small. This leads to a highly sparse matrix of annotations, which may not convey enough information to accurately estimate all the parameters involved in a cumbersome model. This problem is known in the literature as the Cold Start Problem (CSP). The consequence is that simple algorithms like majority voting may perform better in this context.

The first contribution of the thesis is to propose two generative models for the annotation process that exploit the existence of communities of users to alleviate the CSP. Furthermore, identifying the different communities of annotators that exist in a crowdsourcing platform, i.e. spammers, expert annotators, biased annotators, novices annotators, etc., is an worthy goal in itself, as it allows rewarding/removing those annotators. An algorithm that can discover these communities will help the owner of the system to reduce costs by investing the economical resources wisely, or improve the overall performance of the system by addressing the existing biases in the pool of annotators of the system.

The first generative algorithm called the Clustering based Bayesian Combi-

nation of Classifiers (cBCC), assumes that the annotators allocated to the same cluster share the same properties. It relies on the Chinese Restaurant Process (CRP) prior to infer a partition on the annotators. The second algorithm called the Hierarchical Clustering based Bayesian Combination of Classifiers (hcBCC), uses a hierarchical approach to relax the assumption made by the cBCC model. In particular, in the hcBCC the annotators allocated to the same cluster have similar properties, but are not constrained to share the exact same properties. This is more aligned with a practical application in which we expect every person to be different, but groups of people that behave similarly.

We develop efficient Markov Chain Monte Carlo (MCMC) inference algorithms for both models, based on a Gibss sampler with auxiliary variables to deal with non-conjugancies when necessary. The output of this model are the estimates of the ground truth, the properties of the annotators and a partition of the annotators in different clusters.

We evaluate these models with synthetic and real datasets. We vary the level of sparsity of the matrix of annotations to evaluate the robustness of the method to scenarios where the input information is scarce and to analyze how the complexity of the model adapts to the available information. The obtained results are promising, opening a new research challenge in applying BNP tools to develop varying-complexity models for crowdsourcing, in order to address the CSP.

These results can also be found in G. Moreno et al. (2014).

1.2.2 Modelling Inconsistent Annotators

Another goal of using labels coming from multiple annotators is to get very high quality annotated datasets. In this case, we ask several annotators to label nearly all examples in the dataset, and we combine the annotations to get a better estimation of the ground truth. While the underlying concept is the same, the goal is different. In the methods presented previously, the goal was to use crowdsourcing to build a large labeled dataset at a low cost by using unskilled annotators that label a very small part of the dataset. In this case, the primary goal is not to

reduce the cost in terms of time or money, but to maximize the accuracy.

In this situation, the annotation matrix is full and, therefore, we can use more complex models to capture the variability of the performance of the annotators across the instance space. We focus on inductive algorithms which use the features of the instances to estimate not only the ground truth of the training set, but also building a classifier that estimates the ground truth of the unlabeled test instances.

We propose a generative model that identify the different areas across the instance space in which the annotators exhibit a particular behavior. We use a Stick Breaking (SB) to model this heterogeneity in the behavior of the annotators, and we jointly infer a suitable partition of the instance space, the properties of the annotators in each of the clusters of the identified partition, the ground truth of the instances in the training set and a piecewise linear classifier for test instances.

We propose a mean field algorithm relying on a local bound to address non cojungancies in the observation model, and we test the model with synthetic and real datasets.

The results can also be found in G. Moreno et al. (2015).

1.3 Organization

The remainder of this thesis is organized as follows. In Chapter 2, we introduced the topic of crowdsourcing and we review the previous work in this area, making emphasis in the publications related to the problem of aggregating the opinion of several noisy annotators. In Chapter3, we review the basics of BNP models and we focus on their application to Mixture Models (MMs).

The rest of chapters are devoted to our contributions. Chapter 4 introduces the cBCC and hcBCC models, which aim at exploiting the existence of communities of users to alleviate the CSP problem. We proposed MCMC based inference algorithms and we perform experiments to evaluate the performance. In Chapter 5 we present an inductive algorithm to model inconsistent annotators. We proposed a variational inference and evaluate the method using synthetic and real databases. Finally, Chapter 6 is devoted to the conclusions and future research lines.

2

Crowdsourcing

2.1 Introduction

Combining human and machine resources in a symbiotic relationship to solve complex problems is a topic that has received significant attention since the early birth of computers. In fact, Turing (1950) wrote "the idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer". In this line, one of the first goals of artificial intelligence research in the 1950s was to use computers to mimic and even outperform high level human capabilities such as reasoning. By 1960 this goal was quickly reconsidered in light of the first disappointing results (Moravec, 1998).

In his seminal paper, Licklider (1960) envisioned an alternative approach, shifting the focus of research in artificial intelligence from building machines that

outperform humans, to defining a collaborative framework between humans and machines to "enable men and computers to cooperate in making decisions and controlling complex situations without inflexible dependence on predetermined program".

Despite the evolution of computational resources and learning algorithms since then, the truth is that many tasks that are trivial to humans are still a major challenge to machines. Machines exhibit excellent performance in repetitive and mechanistic tasks, or solving problems that involve large scale storage and computing. By the design of adequate machine learning algorithms they can discover complicated patterns in big datasets, or make better decisions that humans when these have to be done based on pure logic. However, humans clearly outperform machines when it comes to high level reasoning, intuition, creativity, social and emotional intelligence among other capabilities.

Recently, Lickliders' vision was retaken giving rise to several prolific research areas like collective intelligence, social computing or human computation. In this context, the concept of crowdsourcing emerged as a distributed problem solving model. The term was coined by Howe (2006), "as the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call". In this way, crowdsourcing can be seen as an alternative to traditional outsourcing, that, with the help of the explosive growth of the Internet offers an attractive and flexible business model for many companies due to:

- The availability of a flexible pool of workers ready to accomplish the work at anytime.
- The easy access to foreign markets eased by the ubiquity of the Internet.
- A flexible payment system based on rewards that achieved a fairly lower price when compared to the price of hiring a dedicated professional through traditional channels.
- Short turnaround time by allowing the task to be distributed among a large

number of wokers.

However, this advantages come at the expenses of some weaknesses that have given rise to an active open research:

- No guarantees about the quality of the work. Unlike traditional approaches where the task was outsourced to a small set of experts, now we are dealing with a large set of non-expert workers for which we have very limited information.

- Lack of confidentiality by exposing the task to a large pool of users over which we do not have control.

This chapter does not intend to provide an exhaustive review of crowdsourcing which, is a diverse and multi-disciplinary field whose taxonomy as well as its definition is still open. Instead, we focus on how to guarantee the quality of the output in a crowdsourcing system by combining the information coming from a large set of non-expert workers for which we have very limited information. Specifically, we tackle the aggregation problem when the workers are solving a multi-class classification task, i.e. the worker is asked to classify some instances in a set of categories identified by a discrete label. It also provide a brief overview of some other main research lines in crowdsourcing for completeness, and complement it with further references for the interested readers.

The rest of the chapter is organized as follows. In Section 2.2, we present some examples of the main categories of crowdsourcing applications. In Section 2.3, we introduce the problem of crowdsourcing aggregation for multi-class classification. Finally, in Section 2.4, we highlight some other related research areas in crowdsourcing.

2.2 Crowdsouring Applications

Crowsourcing is a wide concept whose mere definition is a controversial topic (Estellés-Arolas and González-Ladrón-de Guevara, 2012). This ambiguity translates

into a wide range of crowdsourcing systems whose categorization is not trivial. This section briefly presents some illustrative examples of crowdsourcing systems. For a more exhaustive categorization, the interested reader should refer to Yuen et al. (2011); Nakatsu et al. (2014); Zhao and Zhu (2014) and the references therein.

2.2.1 Knowledge sharing

This category is composed of platforms in which their members share information, generally, in an altruistic way. The members can act as requesters, i.e. consuming or demanding information, or as workers, i.e. providing information. The platform may allow the members to merge their knowledge or edit other members inputs to build the final product. Some prototypical example of this category are:

- *Wikipedia* (Wikimedia Foundation, 2001): an internet-based, open-content encyclopedia created through the collaborative effort of its users which, with more than 4.7 million articles, is the biggest encyclopedia of the world.
- *Youtube* (Hurley et al., 2005): A free video-sharing platform with more than 1 billion users, that allows them to upload and share video clips online.
- *Yahoo Answers* (Yahoo Inc., 2005): A community-driven question-and-answer web site which only in the United States, receives more than 100 million visits per month.

2.2.2 Social games

Also known as implicit crowdsourcing, collecting data from the crowds through games has proven to be efficient and cost-effective. The game keeps users engaged and motivated in solving the task, working as an alternative to economic rewards. Some classic examples of this category are:

- *ESP game* (Von Ahn and Dabbish, 2004): It constitutes the first successful example of harvesting human intelligence through an online game. It addressed the task of image labeling. Once a user entered the system, he was

paired with another random user. A set of images were presented to them, and they had to provide labels. When both of them provided the same label, it become the image label. Later, Google created its own version called Google Image Labeler. Both projects are currently closed.

- *Duolingo* (Von Ahn et al., 2012): It is a free language-learning and crowdsourced text translation platform with more than 10 millions of users. The application allows the users to learn through a question-based game. It also invites users to translate content and vote on translations coming from third party clients.

- *EteRNA* (Lee et al., 2014): The users create sequences of RNA (Ribonucleic Acid), which is an moleculle that control several esential celullar process. To do so, they have to solve different puzzles that are computationally laborious for current computer models. The overall goal of the system is to build a large-scale library of synthetic RNA designs with the contributions of the users.

- r*eCAPTCHA* (Von Ahn et al., 2008): Although not exactly a game, reCAPTCHA is one of the most successful examples of implicit crowdsourcing. Like CAPTCHAs, it asks people to input a text presented to them in the shape of a distorted text image to prevents bots for accessing private areas of a web site. In addition, a second CAPTCHA from old books that cannot be deciphered by computers is shown to the users, who implicitly help to digitize these books.

2.2.3 Marketplaces

This category includes crowdsourcing platforms whose goal is to connect task requesters with task workers. The popularity of these platforms has significantly increased in the last years (see Figure 2.1). We can classify these platforms according to the complexity of the task:

- *Microwork Marketplaces*: The tasks correspond to small pieces of work (microtasks) that the requester distributes to many workers, who complete them at a low price. The most prominent commercial example is Amazon Mechanical Turk (AMT) (Amazon, 2005), which is an online web-based platform where the requesters are able to post tasks known as Human Intelligence Tasks (HITs), e.g. labeling images, translating texts or identifying different descriptions that match the same product. HITs are microtask that are easy for humans but often very hard for computers. AMT offer an Application Programming Interface (API) that allows the requesters to post these HITs in the Marketplace. Workers can then browse among the existing tasks and complete them for a monetary reward set by the requester. Other commercial examples are ClickWorker (Rozsenich et al., 2005) or CrowdFlower (Biewald and Van Pelt, 2007). There also exist non-profitable platforms in which the workers are not motivated by an economic rewards but by other factors. One successful example is Zooniverse (Simpson et al., 2014), which is a crowdsourcing platform that host citizen-based science projects. These projects belong to different categories, e.g. satellite image classification, and the workers are volunteers that contribute to the project because of their personal interest in the topic. Another similar example is (Raimond et al., 2014).

- *Macrowork Marketplaces*: While the microtask marketplaces focus on low complexity task that can be done by almost every worker in less than an hour, macrotask marketplaces focus on more complex tasks that can take longer periods of time and for which certain degree of specialization is required. For example, Elance-oDesk (Elance-oDesk, 2014) offers an online platform in which the requesters can solicit qualified professionals to complete a wide range of specialized tasks in different areas such as design, engineering or technical support. Other examples are LiveOps (Doumar and Feirstein, 2000) which creates virtual call centers with the workers, or Ponoko (Have and Elley, 2007) where the requesters contact qualified designer to cre-

ate customized products from descriptions.

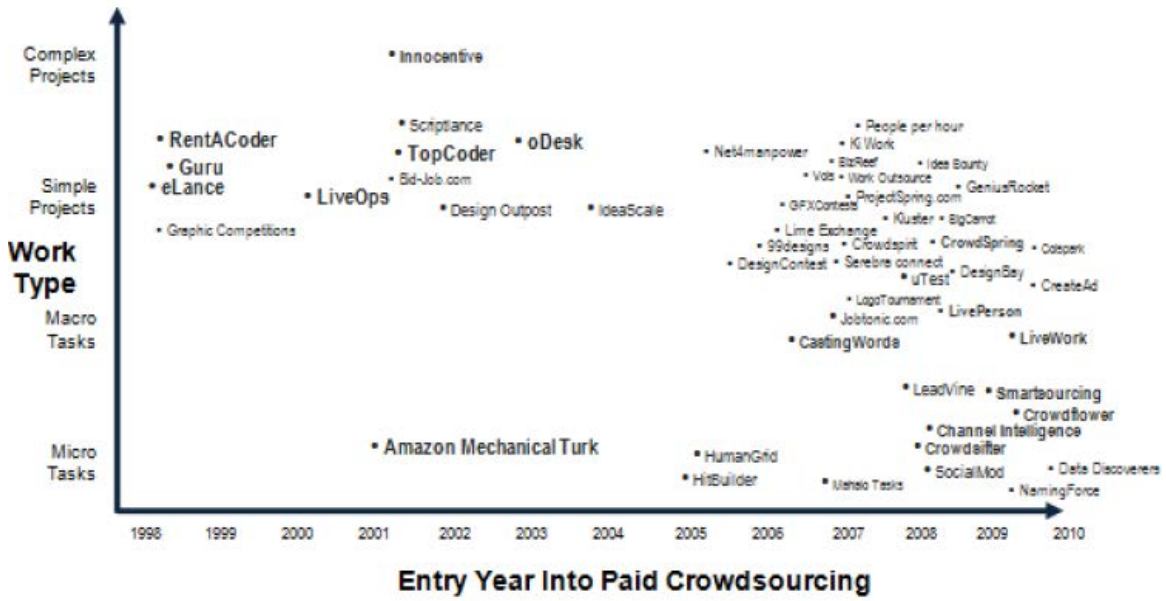


Figure 2.1: Paid Crowdsourcing Vendor Market Entrance. Borrowed from Frei (2009)

As we will see in the next section, this thesis will focus on microwork crowdsourcing platforms and in particular, it will analyze the case when the microtask is a multi-class classification problem. This scenario is especially relevant in Machine Learning, since it offers the possibility of building large labeled datasets in a distributed fashion.

2.3 Aggregating the crowds

One of the main applications of crowdsourcing for machine learning is as a tool to label large datasets to train statistical models. The main advantage is the possibility of distributing the task among a large pool of workers (in this context denoted also as annotators), allowing to complete the job in a short period of time at a low price. However, this comes at a price. Specifically, we delegate the task to a set of annotators that may not be not expert at the task at hand and for which we do not know anything about their level of expertise, their background or their motivations. Therefore, one of the main issues in crowdsourcing systems

is how to aggregate the contributions of a set of potentially unreliable and error-prone sources of information. In particular, we focus on the case in which the task presented to the annotators is a multi-class classification problem, i.e. the annotator has to classify the instance in one of a number of predefined categories, identified by a discrete label.

Let's assume we have a set of instances $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1:N}$ that are labeled by a set of L imperfect annotators for which no information is provided apart from an identification number. The goal is to design an algorithm that receives as input a matrix of annotations $\boldsymbol{Y} \in \{1, C\}^{N \times L}$, where the element $y_{i\ell} \in \{1, C\}$ denotes the label provided by the annotator ℓ for the instance i. This algorithm should output an estimation of the ground truth for the instances $\boldsymbol{Z} = \{z_i\}_{i=1:N}, z_i \in \{1, C\}$, and possibly, an estimation of the ground truth for future unseen instances coming from the same distribution. Notice that the input matrix $\boldsymbol{Y}$ may be sparse since, generally, in practical applications, each anotator only labels a small subset of the total number of instances.

2.3.1 Transductive methods

The earliest methods that we can find in the literature follow a transductive approach. Given the matrix of annotations provided by the annotators for a set of instances, these methods output an estimate of the ground truth for that particular set of instances. However, they are not concerned about how to compute the ground truth for future test instances coming from the same distribution.

The simplest approach to solve the problem is to use a majority voting strategy. This approach relies on the assumption that annotators act independently and without knowing each-other. Moreover, this strategy implicitly assumes that all the annotators have the same performance. This makes the method sub-optimal when there are a few experts and a significant portion of low performance annotators in the pool. In this case, the final estimation is biased toward the opinion of the low performance annotators given that they conform the majority of the population. The situation gets worse if we consider spammers and malicious annotators.

In this case, only a small portion of them can significantly bias the estimation. However, the number of parameters is very low given that all the annotators share the exactly same behavior, which makes majority voting a robust strategy when dealing with highly sparse input matrices. Some variants to provide a more stable solution can be found in the literature. For example, Barowy et al. (2012) propose to use majority voting only when a predefined percentage of agreement among the annotators is reached. In this way, they rule out the possibility that the results are due to random chance.

To alleviate the weaknesses of majority voting, several methods propose to consider the fact that annotators may have different performances due to different factors such as their background or their motivation. One of the simplest approaches consists of including in our training set some additional instances for which the ground truth is known. Based on the answers provided by the annotators for this subset, we can estimate the performance of the different annotators. Finally, based on the estimated properties of the annotators, we can filter those with a low performance (Lee et al., 2010), or we can weight them accordingly to their expertise (Khattak and Salleb-Aouissi, 2011; Snow et al., 2008). These methods can be classified as supervised given that the estimation of the properties of the annotators relies entirely on a subset for which the ground truth is known. The advantage is that it is easy to implement and has a low computational cost. However, it has several drawbacks. First, it is not suitable for problems in which obtaining the ground truth is a costly process. Second, the subset for which the ground truth is observed may not be representative of the problem. A careful design of this labeled subset is needed increasing the cost of the application. For these reasons, this thesis is focused on methods that do not need supervision.

The seminal paper by Dawid and Skene (1979) proposed a latent variable model that takes into account the differences in the performances of the annotators. In addition, it considers that the performance of a particular annotator may depend on the value of the ground truth of the instance. Specifically, the performance of each annotator is represented by a confusion matrix, where each row represents the

conditional probability of the annotator's labeling given the value of the ground truth. In addition, an annotator's labeling process is assumed to be independent of the other annotators given the ground truth. The authors use the Expectation Maximization (EM) algorithm to estimate the Maximum Likelihood (ML) parameters, i.e. the confusion matrices of each annotator and the ground truth, in an unsupervised way. Similar models have been applied to depression diagnosis (Young et al., 1983), myocardial infarction (Rindskopf and Rindskopf, 1986) and image labeling (Smyth et al., 1995), among others.

One of the problems of these methods is the high number of parameters, that can give rise to a low performance when the input matrix of annotations is sparse. Demartini et al. (2012) use a similar model to Dawid and Skene (1979), but instead of modeling each annotator by a confusion matrix, they reduce the number of parameters by using a single scalar, i.e. the performance of the annotators is independent of the ground truth. A similar approach is followed by Liu et al. (2012) and Karger et al. (2011). Kim and Ghahramani (2012) propose a Bayesian extension that allows to include prior information about the characteristics of the annotators and use Gibss sampling to perform inference. They also propose further extension to model the existing correlations among the different annotators. Using this model, Simpson et al. (2011) propose to analyze in a post-processing step the different existing communities of annotators. Venanzi et al. (2014) exploits the existence of those communities to improve the estimates of the ground truth. Wang et al. (2011) published a model in which they distinguish between error rates and biases in an attempt to explicitly model the differences between careful but biased annotators and spammers. Raykar and Yu (2012), following a similar model as Dawid and Skene (1979), proposed to use a prior based on the rank of the confussion matrices of the annotators that is suitable for scenarios with a high proportion of spammers. Hovy et al. (2013) use a binary variable that identify whether an annotator is a spammer, and in this case, it models its labels as coming from a multinomial distribution that is independent on the ground truth.

All the previous methods share a limitation: the annotators are consistent

across the instance space. Several proposals aim at relaxing this assumption by making the properties of the annotators dependent on the difficulty of the instance. Whitehill et al. (2009) propose to use a set of latent variables to model the expertise of the annotators and another set of latent variables to model the difficulty of the instances. A sigmoid function that receives as argument the product of this two latent variables is used to model the probability of the annotator labeling correctly the instance. In this case, the performance of the annotator no longer depends on the ground truth of the instance, but on its difficulty. Also, the model is designed for binary classification problems. A very similar model was proposed by Carpenter (2008). Welinder et al. (2010) extend the previous model by adding a specific bias for each annotator and by considering a high-dimensional concept of difficulty and annotator expertise. One critical aspect of the model is how to select the dimensionality of vector encoding the expertise of the annotator and the vector that encodes the difficulty of the instance. Zhou et al. (2012) set a probability distribution over annotators, instances, and labels. By maximizing the entropy of this distribution, the method naturally infers item confusability and annotators expertise. The ground truth is inferred by minimizing the entropy of this distribution. They proposed a coordinate descent algorithm to optimize the minimax entropy.

While the previous proposals to model the inconsistency of the annotators are more flexible, the number of parameters increase and therefore, they tend to present a poor performance for sparse annotation matrices. To alleviate this problem, Ruvolo et al. (2013) proposed a very similar model to the one proposed by Welinder et al. (2010), but in order to determine commonalities among instances and labelers, they model the latent labeler factors and latent instance factors as linear functions of a specified set of features and an unknown set of weights. The weakness of these methods is that it needs to access a set of observed features that characterized the instances and the annotators. Moreover, for these features to help in learning structural relationships in the data, they have to be carefully chosen.

2.3.2 Inductive methods

While the previous methods are only concerned about estimating the ground truth for the set of instances labeled by the annotators, i.e. $\boldsymbol{Z} = \{z_i\}_{i=1:N}$, the final goal of the methods described in this section is to output a classifier that predicts the ground truth for any test instance coming from the same underlying distribution, i.e. $P(z^*|\boldsymbol{x}^*, \boldsymbol{X})$. We will refer to these methods as inductive methods. Notices that we can use transductive methods to accomplish the same task in a two steps approach. Firstly, we compute from the annotations matrix $\boldsymbol{Y}$ the estimation of the ground truth for the training set $\boldsymbol{Z}$. Secondly, we use any supervised learning algorithm to train a classifier with the labeled set $\{\boldsymbol{x}_i, z_i\}_{i=1:N}$. The idea of inductive methods is that by performing these two steps together, we get an improvement in the overall performance of the system.

Lam and Stork (2005) published one of the first inductive approaches. They proposed a generative model in which an annotator's label is generated only from the ground truth. That is, they assume $P(y_\ell|z, x) = P(y_\ell|z)$. In the same way, the features of the instance $\boldsymbol{x}$ are generated from the ground truth, i.e. $P(\boldsymbol{x}|z)$. When only the annotations and the features are observed, but not the ground truth, the learning problem can be seen as one of learning with missing data, with the ground truth label z being the missing data depending not only on annotations, but also on the features of the data instance itself. A similar approach was published by Carroll et al. (2007). However, neither of both approaches were implemented.

One of the first implementations in this line was published by Raykar et al. (2010). They tackle a binary version of the model published by Dawid and Skene (1979), but modeling the unobserved ground truth as the output of a logistic regression. They used a generalized EM to infer the parameters, i.e. the properties of the annotators, the ground truth of the training set and the parameters of the final logistic regression that can be use to predict the ground truth of instances not yet presented to the annotators. They also proposed a Bayesian extension to incorporate prior knowledge about the properties of the annotators or the distribution of the ground truth. Although the authors claim that the model can be

easily extended to any classifier with a probabilistic output, they just implement a logistic regression. Recently, Rodrigues et al. (2014) published an extension in which the logistic regression was substituted by a Gaussian Process (GP), analyzing the benefits of using more flexible models for the final classifier. They also proposed a new inference scheme based on Expectation Propagation (EP) whose computational cost is similar to the one of training a standard GP for classification. Yan et al. (2012) extended Raykar et al. (2010) to deal with situations in which there is a large set of instances that are not labeled by any of the annotators. To incorporate this information, they model the probability of the ground truth conditioned on the instance by using a graph Laplacian prior.

A slightly different inductive approach is proposed by Rodrigues et al. (2013). Instead of modeling the unknown ground truth labels for which noisy versions are provided by the various annotators, they propose that the focus should be on the annotators themselves. They argue that including the also unknown reliabilities of the annotators as latent variables is preferable, since it not only leads to simpler models that are less prone to overfitting, but also bypasses the problem of the high number of possible labeling to marginalize over. In this way, they use a set of bernoulli latent variables that model the probability of each annotator labelling correctly each of the instances. Unlike the previous methods, the model is derived for a multi-class classification scenario. Another alternative is proposed by Kajino and Kashima (2012). In this case, instead of introducing latent variables to estimate the ground truth, the authors model the labels provided by each annotator using a logistic regressor, and directly relates them to a base classifier that predict the ground truth. This approach leads to a convex optimization problem. They also propose an extension to identify clusters of annotators using a convex clustering penalty (Kajino et al., 2013).

Finally, some recent inductive methods have been published for specific domains. For example Simpson et al. (2015) proposed an extension of Kim and Ghahramani (2012) for document classification in a crowdsourcing scenario. They assume the documents come from a mixture of bag-of-words model, where each

mixture component is a bag-of-words models associated with one particular object class, i.e. conditioned of the ground truth, the words of the documents follow a multinomial distribution over a vocabulary. From there, the annotation process follows the one proposed in Kim and Ghahramani (2012). They proved that by including text-features they achieve better results when the annotations are scarce. An almost identical model was published one year before by Felt et al. (2014).

All the inductive methods mentioned so far, considered the scenario in which the annotators performance is homogeneous across the instance space. Yan et al. (2010) extended Raykar et al. (2010) by modeling the varying precision of each annotator as a function of the instance space. In particular, they use a logistic function of the scalar product between the instance and a vector that parameterized the behavior of the annotator. They infer the Maximum a Posteriori (MAP) value of the parameters using a generalized EM algorithm. Another interesting approach is the one proposed by Zhang and Obradovic (2011). In this case, the authors model the instance space as a Mixture Model (MM). Then, they assume that the annotators have a different behavior in each of the identified components. They use a similar approach to Raykar et al. (2010) to infer the the characteristics of the annotators in those components together with the ground truth and a logistic regression to classify future instances. Another approach to model non-consistent annotators was proposed by Groot et al. (2011). They to assign different variances to the data points for the different annotators, which are then automatically estimated by maximizing the marginal likelihood of the data. However, this last method is restricted to a regression problem.

2.3.3 Related research lines

In active learning scenarios, in addition to our labeled dataset, we have a set of unlabeled data. At each iteration, the algorithm is able to ask to a perfect oracle the true label of a particular unlabeled data. Requesting a label has an associated cost. The goal is maximizing the performance of the learning algorithm while minimizing the incurred cost.

Yan et al. (2011) proposed a pool based active learning algorithm in the presence of multiple noisy labelers. At each iteration, the algorithm try to choose an instance of the pool with large uncertainty, and for which there exists an annotator that can provide a new label with a high level of confidence. They use the inductive model proposed by Yan et al. (2010) and they cast the problem of choosing the instance and the annotator into a bi-convex optimization problem. Rodrigues et al. (2014) model the ground truth as a GP. To select the instance to label, they compute the posterior distribution of the latent variables and they minimize a quantity that provides a balance between the distance to the decision boundary, given by the posterior mean, and the posterior variance (uncertainty) associated with that instance. Regarding the annotator, they choose the annotator that is more likely to label the instance correctly. They also introduce a heuristic to avoid the risk of generating a model that is biased towards labels from a single annotator. Another interesting approach is the one followed in the transductive model proposed by Bachrach et al. (2012). They proposed choosing the label that reduces the most the uncertainty in the estimates of model parameters, as measured by the entropy of the posterior distribution. This approach can be used to choose ground truth labels or annotators' labels with different goals. Specifically, they focus on the case where the ground truth is observed and the goal is to choose which instance should be labeled and which annotator should provided the label, to estimate the annotators' properties as accurately as possible.

Another problem that we find in the literature is how to perform the aggregation when the properties of the annotators vary with the time. Donmez et al. (2010) propose to use a particle filter to model the time-varying accuracies of the different annotators. With the same goal, Simpson et al. (2013) extend their previous model (Simpson et al., 2011) by using a dynamic generalized linear model that allows time-dependent confusion matrices to model the behavior of the annotators.

In this thesis, we work on the problem of aggregating the output of the annotators in a multi-class classification task. However, the problem can be generalized to other scenarios such as soft-label classification Nazabal et al. (2015), multi-label

classification Bragg and Weld (2013), regression Groot et al. (2011) or ranking Wu et al. (2011).

Finally, while the models presented so far have assumed the existence of a objective ground truth, recent publications have started to challenge this assumption. In this line, Wauthier and Jordan (2011) propose a symmetric model in which each annotator is modelled as a linear classifier. The parameters of these linear classifiers are build by adding a set of latent factors shared across the annotators. Instead of assuming a ground truth, the author aims at predicting the labels that one of the annotators, e.g an expert whose opinion we are interested in, would provide for test instances, by integrating out the remaining parameters of the model. Tian and Zhu (2012) tackle the scenario in which each annotator provides labels for different task which may have multiple valid answers, since the task are subjective or poorly specified. They find clusters of annotators in each of the task and based on the sizes of those clusters, they extract a measure of the consistency of each annotator and the subjectivity of each task.

2.4 Other research areas in crowdsourcing

In this section, we briefly summarize some other related research areas in crowdsourcing for completeness. For a more exhaustive review, interested readers should refer to Yuen et al. (2011); Allahbakhsh et al. (2013); Kittur et al. (2013); Zhao and Zhu (2014) and the references therein.

This thesis focuses on combining the information provided by the different workers in a crowdsourcing platform to guarantee the quality of the output. However, this is not the only approach to solve the problem of quality assurance. A prolific related research line aims at analyzing the effect of the remuneration in the quality of the output (Horton and Chilton, 2010; Kazai, 2010) and, accordingly, design effective remuneration policies (DiPalantino and Vojnovic, 2009; Wang et al., 2013; Singer and Mittal, 2013; Difallah et al., 2014). Equally important is understanding the motivations of the workers, which allows us to propose and evaluate the effectiveness of non-financial incentives (Mason and Watts, 2010; Rogstadius

et al., 2011; Kaufmann et al., 2011).

The inclusion of a well-designed credential system in crowdsourcing has a double effect. On one hand, it provides the requesters a way of selecting their workers based on their historical reputation. On the other hand, it has the effect of a non-financial compensation, motivating the workers to correctly perform the job to improve their reputation and improve their chances to be hired again in the future. Multiple papers have researched the role of reputation in crowdsourcing systems and how to use it to design effective pay policies De Alfaro et al. (2011); Zhang and van der Schaar (2012); Allahbakhsh et al. (2013).

Another important factor that directly affects the quality of the output is the definition of the task. Poorly designed task instructions or interfaces can have a serious impact in the quality of the output of the system. Several publications analyze how the task design may limit the performance of the crowd and propose effective design policies to improve the quality or reduce the time to obtain the final result (Kittur et al., 2008; Ipeirotis, 2010; Khanna et al., 2010; Eickhoff and de Vries, 2013; Alagarai Sampath et al., 2014; Alonso, 2013).

In addition to quality assurance, another research line that is getting a lot of attention lately is how to use crowdsourcing in situations in which the work cannot be easily decomposed in a set of microtask, that can be solved independently by a set of unskilled users. Instead, we have several complex tasks for which we may need workers with different levels of specializations and that are highly coupled, e.g. the output of one task may be used to perform another set of related tasks. For example, several papers address the problem of how to decompose a complex task using a hierarchical approach, or how to use collaborative workflows in crowdsourcing platforms Kittur et al. (2011); Kamar and Horvitz (2015); Little et al. (2010); Kulkarni et al. (2012); Dai et al. (2013); Tran-Thanh et al. (2015). An interesting related research line is tasks allocation in crowdsourcing systems with complex workflows under a budget constraint Tran-Thanh et al. (2014, 2015).

Real-time crowdsourcing is also becoming popular. The main challenge is how to optimally allocate tasks to workers to achieve successful completion of the tasks

under real-time constraints. With this goal, different proposals analyze how to stimulate user participation and handle dynamic task assignment, in such a way so that the real-time demands are met, and high quality results are delivered (Boutsis and Kalogeraki, 2013; Bernstein et al., 2012; Lasecki and Bigham, 2013).

3

Bayesian Nonparametrics

3.1 Introduction

Most of the traditional machine learning algorithms split the learning task into two sub-problems. Firstly, given a family of models indexed by a set of finite-dimensional parameters θ, how to infer the value of the parameters from the training set $\mathcal{D}$. And secondly, how to determine the family of models to consider in the first place.

The second problem is known in the literature as model selection and it is crucial to guarantee that the final model faithfully solves the learning task. In particular, if we consider a too large family of models, we will always be able to find one that perfectly captures our training set, but that would be unable to generalize for future unseen data, i.e. the algorithm will overfit to the training set. On the contrary, if the family of models is too small, no matter the inference algorithm

we use, we would end up with a model that is not flexible enough to capture the underlying distribution that generated the data and we would incur in what is called underfitting. Some typical examples of model selection are choosing the number of components in a Finite Mixture Model (FMM), choosing the number of hidden states in a Hidden Markov model (HMM) or selecting the number of features in a Factor Model (FM).

A different approach is to consider a family of models whose complexity can grow with the size of the training set. Nonparametric algorithms follow this approach. Some classical examples that we can find in the literature are K-Nearest Neighbor, Support Vector Machines or Parzen Window estimators.

Bayesian Nonparametric (BNP) models follow the approach of nonparametric algorithms inside a Bayesian framework. Specifically, to allow the number of parameters of our model to be unbounded and to depend on the amount of data we observe, we need to assume an infinite number of parameters a priori. After observing a finite amount of data we expect that only a finite subset of these parameters is used. In this way, BNP models offer an alternative to traditional model selection techniques, considering a family of models of varying complexity, and inferring the parameters and the complexity from the data instead of tackling the problem in two separate stages.

In this chapter we present an overview of the BNP concepts that are used in the rest of the thesis. We start by reviewing the main ideas of Bayesian statistics in Section 3.2, and we explain in more detail Bayesian Mixture Model (BMM) as one of the building blocks of the algorithms presented in Chapters 4 and 5. In Section 3.3, we review the theory of BNP models. Specifically, we focus on the Dirichlet Process (DP) as a prior of Dirichlet Process Mixture Model (DPMM).

3.2 Bayesian Models

In this section we review the main ideas behind Bayesian statistics as it will be the mathematical tool to express and support the algorithms proposed in this thesis.

The philosophical battle between those in favor of Bayesian statistics, and those

in favor of a frequentist approach is far from being a closed discussion, and we are not in position to give a final answer in this document. We will limit this section to expose some of the reasons that justify the use of Bayesian statistics, both as a mathematical tool based on strong axiomatic foundations which guarantee the mutual consistency of the methods proposed, and as a mature field that allows, from a practical point of view, to construct and manipulate complex models based on a simple set of well defined rules.

The axiomatic definition of probability theory (Kolmogorov, 1950; Renyi, 1970) was designed to provide a measure-theoretic probability calculus, i.e. a definition of the rules for constructing and manipulating mathematical statements involving probabilities. Unfortunately, this axiomatization only tells us how to manipulate probabilities, i.e. it does not tell us what they are or how to interpret them.

At the center of Bayesian statistics is the need of describing by means of a probability distribution all the uncertainties that we encounter in our problem. In that sense, like any other statistical analysis, it relies on the specification of a probability model, which is the underlying mechanism that generates the observed data $\mathcal{D} = \{\boldsymbol{x}\}_{i=1:N}$. This probability model is described by a function of a parameter $\theta \in \Theta$. The particularity of Bayesian statistics is the treatment of θ as a random variable. This does not model the variability of θ, which is a parameter and therefore a fixed quantity. Instead, the distribution $P(\theta)$ captures the uncertainty we have about its value before observing the data $\mathcal{D}$.

Under this framework, the learning process is given by the Bayes' theorem, that can be seen as a transformation that maps $P(\theta)$, i.e. the uncertainty about θ before observing the data, into $P(\theta|\mathcal{D})$, i.e. the uncertainty about θ after observing the data:

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{\int_\Theta P(\mathcal{D}|\theta)dP(\theta)}, \tag{3.1}$$

and this in turn provides information about the distribution of future data $\boldsymbol{x}$ generated from the same probabilistic model, in which the posterior uncertainty is integrated out:

$$P(\boldsymbol{x}|\mathcal{D}) = \int P(\boldsymbol{x}|\theta, \mathcal{D})dP(\theta|\mathcal{D}). \tag{3.2}$$

It is important to notice that the prior $P(\theta)$ encodes the information we have about the fixed value θ before observing any data. The likelihood term $P(\mathcal{D}|\theta)$ encodes our probability model. As number of data $n \to \infty$, the likelihood dominates and the prior term vanishes. This is not the same as saying that the model is consistent, i.e. as $n \to \infty$ the posterior does not necessarily concentrate around the true value θ. Consistency would need further assumptions regarding the correctness of the probability model or the support of the prior.

In Section 3.3, we justify this framework under the mild assumption of exchangeability and we use this to motivate the need of BNP models. But before that, we will apply this theory to Mixture Models (MMs) as they constitute an important element that will repeatedly appear in the rest of this document. For more information about the Bayesian theory and the philosophy behind it, we refer the interested readers to Jaynes (2003).

3.2.1 Bayesian Mixture Models

Since the seminal work of Pearson (1894), FMMs have been used in a wide range of different disciplines such as astronomy, biology, engineering, genetics, medicine, social sciences and so on. FMMs can be easily applied to datasets in which two or more sub-populations are mixed together. Due to its flexibility in modeling, FMMs have enjoyed intensive attention over the past years, from both practical and theoretical viewpoints.

Specifically, FMMs constitute an statistical framework in which each data is assumed to come from one out of K groups. The distribution function of each group is referred to as a component of the FMM. These components are weighted by the relative frequency of the components in the population.

Specifically, assuming an observed fixed number of components K, we model a dataset $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1:N}$ as a collection of i.i.d. draws from a mixture distribution

$$\boldsymbol{x}_i|\boldsymbol{\phi} \sim \sum_{k=1}^{K} \pi_k P(\boldsymbol{x}_i|\boldsymbol{\phi}_k), \tag{3.3}$$

where $P(\boldsymbol{x}|\boldsymbol{\phi}_k)$ is a given parametric family of distributions indexed by a parameter

$\boldsymbol{\phi}_k \in \Phi$ and $\boldsymbol{\pi} \in \mathcal{S}^K$ are the mixture proportions, where $\mathcal{S}^K$ denotes the K-dimensional probability simplex.

An equivalent and more practical formulation of this model from the inference perspective can be derived by assuming a collection of latent allocation random variables. In particular, we assume that each observation $\boldsymbol{x}_i$ is generated from a specific but unknown component $q_i \in \{1 \dots K\}$. We can then rewrite Equation 3.3 in terms of a collection of auxiliary random variables $\boldsymbol{q} = \{q_i\}_{i=1:N}$ that are i.i.d. with $P(q_i = k) = \pi_k$. The generative model is the following:

$$\boldsymbol{x}_i | q_i, \boldsymbol{\phi} \sim P(\boldsymbol{\phi}_{q_i}),$$
$$q_i | \boldsymbol{\pi} \sim \text{Discrete}(\boldsymbol{\pi}).$$

Notice that integrating out the random variables $\{q_i\}_{i=1:N}$ we recover again Equation 3.3.

Finally, in a Bayesian setting (see Section 3.2) we model the uncertainty about the unobserved quantities that we want to infer by posing prior distributions on them. In this case we need to define a suitable prior on the parameters of the different components, as well as a prior on the weights on the components. For the latter, a common choice is to use a Dirichlet distribution. The choice of the prior on the parameters of the components depends on the model we use to represent the observations,

$$\boldsymbol{\pi} | \boldsymbol{\alpha} \sim \text{Dir}(\boldsymbol{\alpha}), \tag{3.4}$$
$$\boldsymbol{\phi}_k | \boldsymbol{h} \sim P(\boldsymbol{h}), \tag{3.5}$$

where $\boldsymbol{\alpha} \in \mathbb{R}_+^{1 \times K}$ is a vector of real positive values. A common choice is to use a symmetric Dirichlet distribution, i.e. $\alpha_i = \alpha_j, \forall i, j$. The collection of random variables $\boldsymbol{\phi} = \{\boldsymbol{\phi}_k\}_{k=1:K}$ and $\boldsymbol{h}$ are vectors of parameters and hyper-parameters respectively, whose support depends on the particular modeling problem.

Putting this altogether, we have a BMM whose graphical representation can be seen in Figure 3.1.

This generative process yields the following joint distribution over the observed

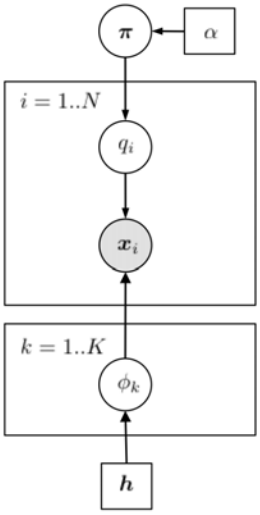


Figure 3.1: Graphical model for a generic Bayesian Mixture Model

data, the latent allocation variables and the parameters of the components:

$$P(\mathcal{D}, \boldsymbol{q}, \boldsymbol{\phi}|\boldsymbol{\alpha}, \boldsymbol{h}) = \prod_{k=1}^{K} \left[\left(\prod_{i:q_i=k} P(\boldsymbol{x}_i|\boldsymbol{\phi}_k) \right) \pi_k^{\sum_{i=1}^{N} \mathbb{I}(q_i=k)} P(\boldsymbol{\phi}_k|\boldsymbol{h}) \right] P(\boldsymbol{q}), \quad (3.6)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

Given an observed dataset, and from a clustering perspective, we are interested in finding a partition of the data in K different clusters. Applying the Bayes theorem, we can compute the posterior probability of the latent variables $\boldsymbol{q}$ given the data $\mathcal{D}$:

$$P(\boldsymbol{q}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{h}) = \frac{P(\mathcal{D}|\boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{h}) P(\boldsymbol{q}|\boldsymbol{\alpha}, \boldsymbol{h})}{\sum_{\boldsymbol{q}} P(\mathcal{D}|\boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{h}) P(\boldsymbol{q}|\boldsymbol{\alpha}, \boldsymbol{h})}, \quad (3.7)$$

where the likelihood $P(\mathcal{D}|\boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{h})$ can be obtained by integrating out the parameters of the components $\boldsymbol{\phi}$:

$$P(\mathcal{D}|\boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{h}) = \int_{\Phi} \prod_{k=1}^{K} \left[\left(\prod_{i:q_i=k} P(\boldsymbol{x}_i|\boldsymbol{\phi}_k) \right) dP(\boldsymbol{\phi}_k|\boldsymbol{h}) \right]. \quad (3.8)$$

If $p(\phi_k|\boldsymbol{h})$ is the conjugate prior of $p(\boldsymbol{x}_i|\boldsymbol{\phi}_k)$, then the integral in Equation 3.8 can be computed analytically. Finally, $p(\boldsymbol{q}|\boldsymbol{\alpha}, \boldsymbol{h})$ is obtained by marginalizing the weights $\boldsymbol{\pi}$:

$$p(\boldsymbol{q}|\boldsymbol{\alpha}, \boldsymbol{h}) = \int_{\mathcal{S}^K} \prod_{k=1}^{K} \left[\pi_k^{\sum_{i=1}^{N} \mathbb{I}(q_i=k)} \right] \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}).d\boldsymbol{\pi} \quad (3.9)$$

In this case, the integral in Equation 3.9 can be computed analytically given that the Dirichlet distribution is the conjugate prior to the Multinomial distribution. In particular, the distribution $p(\boldsymbol{q}|\boldsymbol{\alpha}, \boldsymbol{h})$ follows a Dirichlet Compound Multinomial Distribution:

$$p(\boldsymbol{q}|\boldsymbol{\alpha}, \boldsymbol{h}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Gamma(N + \sum_{k=1}^{K} \alpha_k)} \prod_{k=1}^{K} \frac{\Gamma(\sum_{i=1}^{N} \mathbb{I}(q_i = k) + \alpha_k)}{\Gamma(\alpha_k)}, \tag{3.10}$$

where $\Gamma(\cdot)$ denotes the Gamma function. The posterior over the latent allocation variables defined in Equation 3.7 is intractable because the marginalization in the denominator involves summing over every partition of the data in K clusters. This sum has a number of terms given by the Stirling's number of the second kind (Gould, 1960), growing exponentially with the number of data. This makes necessary to resort to approximate inference techniques (see Section 3.3.5).

3.3 Bayesian Nonparametric Models

In this section we briefly justify the use of BNP models using the concept of exchangeability. In particular, we will focus on the DP and its use as a prior for DPMM.

3.3.1 Exchangeability

One way of justifying the use of Bayesian statistics from a mathematical point of view relies on the concept of exchangeability, which plays a crucial role in motivating the development of Bayesian analysis in general and BNP models in particular.

Suppose we have an indexed sequence of data $\{\boldsymbol{x}_i\}_{i=1:N}$ with $\boldsymbol{x}_i \in \mathcal{X}$, and let's g be a finite permutation of $[N]$, i.e. the set of integers $1, 2, \ldots N$.

Then we say the sequence $\{\boldsymbol{x}_i\}_{i=1:N}$ is exchangeable if

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) = P(\boldsymbol{x}_{g(1)}, \boldsymbol{x}_{g(2)}, \ldots, \boldsymbol{x}_{g(N)}), \quad \forall g \in \mathbb{G}^N, \tag{3.11}$$

where $\mathbb{G}^N$ is the symmetric group of $[N]$, i.e. the group of all permutations of the set $[N]$.

Exchangeability is a mild assumption that is fulfilled in a wide range of statistical problems in which the information carried by data is independent of the order in which they are collected. Notice that independency implies exchageability but the reverse is not true.

We are interested in sequences that are exchangeable for any $N \in \mathbb{N}$ as it is logical to assume that the exchangeability assumption holds not only for our training sequence, but for any test sequence over which we want to make predictions.

We define an infinitely exchangeable sequence as a random sequence in which any subset is finitely exchangeable. More formally

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\infty) = P(\boldsymbol{x}_{g(1)}, \boldsymbol{x}_{g(2)}, \ldots, \boldsymbol{x}_{g(\infty)}), \quad \forall g \in \mathbb{G}^\infty \tag{3.12}$$

, where $\mathbb{G}^\infty$ is an infinite symmetric group.

A key results that follows from the infinitely exchangeable assumption is given by the Representation Theorem published by Bruno de Finetti in 1937 (See de Finetti (1980) for an English translation) and its posterior generalization by Hewitt and Savage (1955) and Ryll-Nardzewski (1957). Given an infinitely exchangeable sequence $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\infty)$ of arbitrary random quantities $\boldsymbol{x}_i \in \mathcal{X}$ with joint probability distribution $P(\cdot)$, this theorem states that there exists an integral representation of the form:

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) = \int_\Theta \prod_{i=1}^{N} P(\boldsymbol{x}_i|\theta) dP(\theta), \tag{3.13}$$

where $P(\boldsymbol{x}_i|\theta)$ is a probabilistic model parameterized by $\theta \in \Theta$. Moreover, θ is the limit as $n \to \infty$ of some function of the observations.

Roughly speaking it tells you that any subset of this sequence is a random sample from a probabilistic model and that there exists a prior which describes the uncertainty about the parameter θ before observing the data.

Notices that while this theorem justifies the use of Bayesian statistics, it does not tell anything about the underlying probability model. This model may be arbitrarily complex, and may even require an infinite-dimensional parameterization θ to describe it, justifying the need of BNP models to avoid the often restrictive assumptions of parametric models.

When dealing with an infinite number of parameters we no longer talk about probability density functions but about stochastic processes. Understanding how to manipulate these mathematical objects is crucial to understand the theory behind BNP models. The dramatic advances in the implementation of approximate inference algorithms in the last years made BNP models not only an elegant theoretical framework, but also a practical tool directly applicable to a wide range of real-world problems. In the rest of this section we will focus on BNP models for BMM.

3.3.2 Dirichlet Process

3.3.2.1 Definition

The DP is a stochastic process whose realizations are random infinite discrete probability distributions (Ferguson, 1973). Let Θ be a measurable space, then a DP is completely specified by a base distribution G_0 on Θ (which is the expected value of the process) and a positive real number α (usually referred to as concentration parameter) which plays the role of an inverse variance.

As other stochastic processes, a DP can be characterized by its weak distribution[1]. In particular, let G_0 be a probability measure on a measurable space Θ and $\alpha \in \mathbb{R}_+$. Then, a random probability measure G over Θ constitutes a DP if its measure on any finite measurable partition $(A_1, A_2, \ldots, A_r)$ of Θ, i.e. a set of disjoint measurable subsets whose union equal to Θ, is distributed as a finite-dimensional Dirichlet distribution of the form

$$(G(A_1), G(A_2), \ldots, G(A_r)) \sim \mathrm{Dir}\left(\alpha G_0(A_1), \alpha G_0(A_2), \ldots, \alpha G_0(A_r)\right). \tag{3.14}$$

Moreover, for a given G_0 and α there is a unique stochastic process satisfying the stated conditions.

We write $G \sim \mathrm{DP}(\alpha, G_0)$ if G is a random probability measure with distribution given by the DP. The first two cumulants of the DP are given by

$$\mathbb{E}[G(A)] = G_0(A) \tag{3.15}$$

[1]The weak distribution of a stochastic process is the set of all its finite-dimensional marginals.

and

$$\text{Var}[G(A)] = \frac{G_0(A)(1 - G_0(A))}{\alpha + 1}. \tag{3.16}$$

For any measurable set $A \in \Theta$ we have:

$$\lim_{\alpha \to \infty} G(A) = G_0(A). \tag{3.17}$$

So G converges to G_0 weakly or pointwise. Notice that this is not equivalent to saying that G converges to G_0. In Section 3.3.2.2, we will show that the distributions corresponding to draws of a DP are discrete with probability one, even when G_0 is non atomic.

3.3.2.2 Posterior Distribution

In the same way that the Dirichlet distribution is the conjugate prior to the Multinomial distribution, i.e., the posterior distribution of a Multinomial likelihood with a Dirichlet prior is again a Dirichlet distribution, it can be shown that the DP is also conjugate for estimating a completely unknown distribution from i.i.d. instances. In particular, let $G \sim \text{DP}(\alpha, G_0)$ be a random measure distributed according to a DP and let $\theta_i \sim G$ be N i.i.d. draws from G. Ferguson (1973) showed that the posterior measure follows a DP as well:

$$P(G|\{\theta_i\}_{i=1:N}, \alpha, G_0) = DP\left(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_\theta(\theta_i)\right), \tag{3.18}$$

where $\delta(\cdot)$ is the Dirac delta function. This can be used to show that the samples from a DP are discrete distributions with probability one. Specifically, for every measurable set $A \in \Theta$, from Equations 3.18 and 3.15 we have

$$\mathbb{E}\{G(A)|\{\theta_i\}_{i=1:N}, \alpha, G_0\} = \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}(A), \tag{3.19}$$

which is a convex combination of the prior mean G_0 and the empirical distribution, where the weight of the prior mean is proportional to the concentration parameter α and the weight given to the empirical distribution is proportional to the number of samples. If we take the limit when $N \to \infty$, the first term of the right hand

side vanishes and we end up with the following expression:

$$\lim_{n\to\infty} \mathbb{E}\{G(A)|\{\theta_i\}_{i=1:N}, \alpha, G_0\} = \sum_{k=1}^{\infty} \pi_k \delta_\theta(\boldsymbol{\phi}_k), \tag{3.20}$$

where $\{\boldsymbol{\phi}_k\}_{k=1:\infty}$ are the unique values in the original sequence $\{\theta_i\}_{i=1:\infty}$, and π_k is the empirical frequency of those values. If we assume that the posterior concentrates around its mean, then the posterior is purely atomic which implies that the realizations of the DP are discrete with probability one. A formal proof was provided by Ferguson (1973); Blackwell (1973); Blackwell and Macqueen (1973) and subsequently by Sethuraman (1994). The importance of the discrete nature of the observations of the DP is that it is able to generate ties among the observations, making the DP a very suitable prior for clustering applications (see Section 3.3.4).

3.3.2.3 Predictive Distribution

Given the following generative process,

$$\theta_i \sim G, \quad \forall i,$$
$$G \sim DP(\alpha, G_0),$$

we want to compute the predictive distribution of θ_{N+1} after observing $\{\theta_i\}_{1:N}$. Notice that the observations $\{\theta_i\}_{1:N+1}$ are conditionally independent given G, so for every measurable set $A \in \Theta$ we have:

$$P(\theta_{N+1} \in A|\{\theta_i\}_{1:N}) = \mathbb{E}\{G(A)|\{\theta_i\}_{1:N}\} = \frac{\alpha}{\alpha+N} G_0(A) + \frac{1}{\alpha+N} \sum_{i=1}^{N} \delta_\theta(A). \tag{3.21}$$

Taking into account that this holds for every measurable set $A \in \Theta$, we obtain the following final expression for the predictive distribution:

$$\theta_{N+1}|\{\theta_i\}_{1:N} \sim \frac{\alpha}{\alpha+N} G_0 + \frac{1}{\alpha+N} \sum_{i=1}^{N} \delta_\theta(\theta_i), \tag{3.22}$$

so the predictive distribution $\theta_{N+1} \in A|\{\theta_i\}_{1:N}$ is equal to the posterior base distribution derived in Section 3.3.2.2. For a formal proof see Blackwell and Macqueen (1973).

Equation 3.22 provides a method to draw samples from an unobserved random measure $G \sim DP(\alpha, G_0)$, without explicitly constructing G. This generative process can be seen as a *Pólya Urn* scheme (Eggenberger and Pólya, 1923). Let us assume that each $\theta \in \Theta$ represents a unique color. We draw balls whose colors are given by $\theta \sim G$. In addition, assume that we have an urn containing the balls associated with previous draws θ. At the beginning the urn is empty, so we draw the new ball and we paint it with a color sampled from the base distribution $\theta_1 \sim G_0$. For subsequent draws, with probability $\frac{n}{\alpha+n}$ we draw a ball from the urn that contains the n balls drawn previously, we paint the new ball with the same color and drop both balls into the urn again $\theta_{n+1} \sim \sum_{i=1}^{n} \delta_\theta(\theta_i)$. However, with probability $\frac{\alpha}{\alpha+n}$ we paint the new ball with a previously unseen color $\theta_{n+1} \sim G_0$.

Denoting by $\{\boldsymbol{\phi}_k\}_{i=1:K}$ the unique values in the original sequence $\{\theta_i\}_{i=1:N}$, we can rewrite the predictive distribution in the following way:

$$\theta_{N+1}|\{\theta_i\}_{1:N} \sim \frac{\alpha}{\alpha+N} G_0 + \frac{1}{\alpha+N} \sum_{k=1}^{K} n_k \delta_\theta(\phi_k), \tag{3.23}$$

so θ_n will take a previously seen value ϕ_k with probability proportional to the number of times it has already been observed n_k. The larger n_k is, the larger the probability that the cluster k grows. This is a property informally called rich-gets-richer and leads to the fact that draws $G \sim DP(\alpha, G_0)$ are discrete with probability one (Blackwell and Macqueen, 1973).

3.3.2.4 Stick Breaking Construction

In Section 3.3.2.1, we define a DP as a random probability measure that we characterize in terms of its weak distribution. This is an implicit description since it does not tell us how to draw a realization from a DP. So far, we only know that if a measure G is a realization from a $DP(\alpha, G_0)$, then it is discrete with probability one, so it admits the following representation:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_\theta(\phi_k). \tag{3.24}$$

However, the proofs provided by Ferguson (1973); Blackwell (1973); Blackwell and Macqueen (1973) are not constructive, i.e. they do not provide a way to generate

the parameters $\{\pi_k, \phi_k\}_{k=1:\infty}$ of this representation.

Sethuraman (1994) provided an alternative proof of the discreteness of the realizations of a DP that relies on an explicit construction of the the random measure $G \sim DP(\alpha, G_0)$, called the Stick Breaking (SB) construction.

More formally, given a random infinite sequence $\{\pi_k\}_{k=1:\infty}$, a set of auxiliary i.i.d. variables $\{v_k\}_{k=1:\infty}$ and a positive parameter α, we define a SB process in the following way:

$$v_k \sim \text{Beta}(1, \alpha), \quad k = 1, 2, \ldots, \tag{3.25}$$

$$\pi_k = v_k \prod_{\ell=1}^{k-1} (1 - v_\ell). \tag{3.26}$$

The sequence $\{\pi_k\}_{k=1:\infty}$ defines a valid probability mass function as $\sum_{k=1}^{\infty} \pi_k = 1$ with probability one, given that

$$1 - \sum_{k=1}^{K} \pi_k = \prod_{k=1}^{K} (1 - v_k). \tag{3.27}$$

We write $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$ if $\boldsymbol{\pi}$ is a random probability measure over the set of positive integers $\mathbb{N}_+$ and it is generated by Equations 3.25 and 3.26 (GEM stands for Griffiths, Engen and McCloskey) (Pitman, 2002).

We can understand the construction of the sequence $\{\pi_k\}_{k=1:\infty}$ in the following recursive way. Starting with a stick of unit length, at each iteration $k = 1, 2, \ldots$ a piece of relative length v_k is broken off (relative to the current length of the stick). See Figure 3.2 for an illustration.

Sethuraman (1994) proved that if we have a random measure G on a measurable set Θ that is generated in the following way:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_\theta(\phi_k), \quad \phi_k \sim G_0, \tag{3.28}$$

where G_0 is a measure on Θ, $\alpha \in \mathbb{R}_+$ and $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$, then $G \sim DP(\alpha, G_0)$. Conversely, samples from a DP are discrete with probability one and have a representation given by Equation 3.28.

To gain some intuition about why that is the case, assume that we draw an observation $\theta_1 \sim G$ where $G \sim DP(\alpha, G_0)$. Then, the posterior distribution of

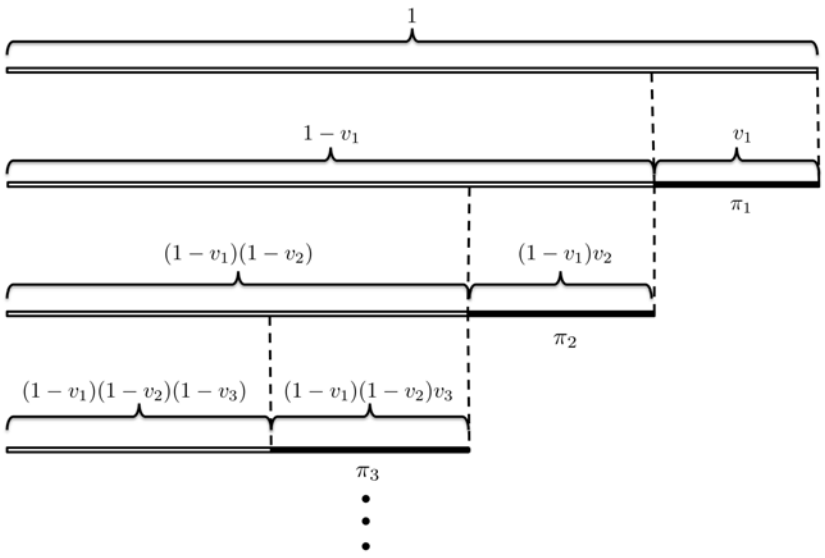


Figure 3.2: Illustration of the stick-breaking construction for the DP. From the initial stick with length equal to 1 we remove a portion with random length $v_1 \sim \text{Beta}(1, \alpha)$ and we assign its length to the first sample π_1. From the remaining stick of length $1 - v_1$ we remove a new portion with random length $v_2(1 - v_1)$ where $v_2 \sim \text{Beta}(1, \alpha)$ and we assign its length to the second sample π_1. Note that π_k do not monotonically decrease.

$G|\theta$ is given by:

$$P(G|\theta_1, \alpha, G_0) = DP\left(\alpha + 1, \frac{\alpha}{\alpha + 1} G_0 + \frac{1}{\alpha + 1}\delta_\theta(\theta_1)\right), \tag{3.29}$$

and applying the definition given in Section 3.3.2.1, $G(\{\theta_1\}) \sim \text{Beta}(1, \alpha)$, assuming that G_0 is non-atomic. The remaining probability is spread over the complementary set $\{\theta_1\}^c$ and $G(\{\theta_1\}|\{\theta_1\}^c) \sim DP(\alpha, G_0)$ independently of $G(\{\theta_1\})$. Therefore,

$$G = Y\delta_\theta(\theta_1) + (1 - Y)G, \tag{3.30}$$

$$Y \sim \text{Beta}(1, \alpha). \tag{3.31}$$

It can be proved the the solution to this equation is unique, and it is straightforward to check that the solution is given by the SB representation.

Finally, we can relate the parameter α with the mean value of the auxiliary

variables $\{v\}_{k=1:\infty}$:

$$\mathbb{E}\{v_k\} = \frac{1}{1+\alpha}. \tag{3.32}$$

Therefore, for small values of α almost all the probability mass is distributed among the first few components. As $\alpha \to \infty$, G weakly converge to the base distribution G_0, assigning roughtly uniform weights to a dense set of discrete parameters $\{\theta_k\}_{k=1:\infty}$.

The importance of the SB representation lies in the fact that, along with an appropriate truncation stage, it allows us to generate a DP approximately. There are several results about the accuracy of this approximation for a given choice of a concentration parameter α, a sample size N and a truncation level T (Ishwaran and James, 2001; Blei and Jordan, 2006). Moreover, this representation will be used for the majority of the algorithms that resort to variational approximations to infer the posterior of a DPMM (See Section 3.3.4). Finally, this representation allows building new random measures by changing the distribution of $v_k \sim \text{Beta}(1, \alpha)$ to other possibilities (Pitman, 2002; Pitman and Yor, 1997).

3.3.3 Chinese Restaurant Process

Suppose a finite set of observations $\{\theta_i\}_{i=1:N}$ from a random measure $G \sim DP(\alpha, G_0)$. The discreteness of G creates ties among the elements of the set $\{\theta_i\}_{i=1:N}$, so the set of unique values $\{\boldsymbol{\phi}_i\}_{i=1:K} \subseteq \{\theta_i\}_{i=1:N}$. Therefore, an equivalent representation of the random variables $\{\theta_i\}_{i=1:N}$ can be given in terms of random partitions of the set $[N] = \{1, 2, \dots N\}$. Let $\widehat{\boldsymbol{\pi}}_N$ be a random partition of the set $[N]$, i.e. a family of random subsets of $[N]$ such that i and i' belong to the same subset or cluster if and only if $\theta_i = \theta_j$. Let us denote by $\boldsymbol{\phi}_k$ the unique value associated with cluster $k \in \widehat{\boldsymbol{\pi}}_N$, and by $\mathcal{P}_N$ the space of all possible partitions $\widehat{\boldsymbol{\pi}}_N \in \mathcal{P}_N$.

Likewise, an infinite set of observations $\{\theta_i\}_{i=1:\infty}$ induces a partition on the set $\mathbb{N}_+$ denoted by $\widehat{\boldsymbol{\pi}}_\infty \in \mathcal{P}_\infty$. Given that the sequence $\{\theta_i\}_{i=1:\infty}$ is exchangeable, the partition $\widehat{\boldsymbol{\pi}}_\infty \in \mathcal{P}_\infty$ is also exchangeable. Therefore, by de Finetti's Theorem, there exists a latent random probability measure such that, conditioned on it, the observations become independent. Such a random measure is the SB process.

Moreover, if we denote by $q_i = k$ the event that i belongs to the cluster k, the predictive distribution is given by the Chinese Restaurant Process (CRP):

$$p(q_N = k|\widehat{\boldsymbol{\pi}}^{\neg N}, \alpha) \propto \begin{cases} |k|^{\neg N}, & k \in \widehat{\boldsymbol{\pi}}^{\neg N} \\ \alpha, & k = \emptyset, \end{cases} \tag{3.33}$$

where $\widehat{\boldsymbol{\pi}}^{\neg N}$ denotes the partition given by the previous assignments, i.e. $\{q_i\}_{i=1:N}$. $|k|$ represents the cardinality of cluster k and $|k|^{\neg N}$ is equal to $|k|$ excluding the element N. Finally, $q_N = \emptyset$ denotes the event that the element N creates a new singleton cluster.

Alternatively, this process can also be viewed as the CRP culinary metaphor. Assume that we have a Chinese Restaurant with an infinite number of tables, each of them with a capacity for an infinite number of customers. Suppose that the first customer arrives and sits at a randomly chosen empty table. When a second customer arrives at the restaurant he can either occupy the table in which the first customer is allocated or he can sit at another table. In general, the N-th customer sits at an already occupied table k with a probability proportional to the number of customers already allocated to that table, $|k|$, or he sits at a new table with probability proportional to a positive real parameter α. If we identify customers with elements in the set $\mathbb{N}$ and tables with clusters, after N customers has enter the restaurant the tables define a partition $\widehat{\boldsymbol{\pi}}_N \in \mathcal{P}_N$ whose distribution is given by the CRP. See Figure 3.3 for a sketch of the CRP.

It is important to remark that a partition $\widehat{\boldsymbol{\pi}}_\infty \sim \text{CRP}(\alpha)$ is infinitely exchangeable, so the probability mass function of any finite partition $\widehat{\boldsymbol{\pi}}_N$ given by the restriction of $\widehat{\boldsymbol{\pi}}_\infty$ to the set $[N]$ depends only on the number of clusters $K = |\widehat{\boldsymbol{\pi}}_N|$ and the sizes of the clusters $\{|k|\}_{k=1:K}$. Specifically, we have:

$$\widehat{\boldsymbol{\pi}}_N \sim \frac{\Gamma(\alpha)\alpha^{|\widehat{\boldsymbol{\pi}}_N|}}{\Gamma(\alpha + N)} \prod_{k \in \widehat{\boldsymbol{\pi}}_N} \Gamma(|k|). \tag{3.34}$$

Finally, it is interesting to analyze the expected number of tables K after N costumers arrived at the restaurant. In particular, notice that for $i \geq 1$ the probability that the customer i sits at a new table is $\alpha/(\alpha + i - 1)$, so the average

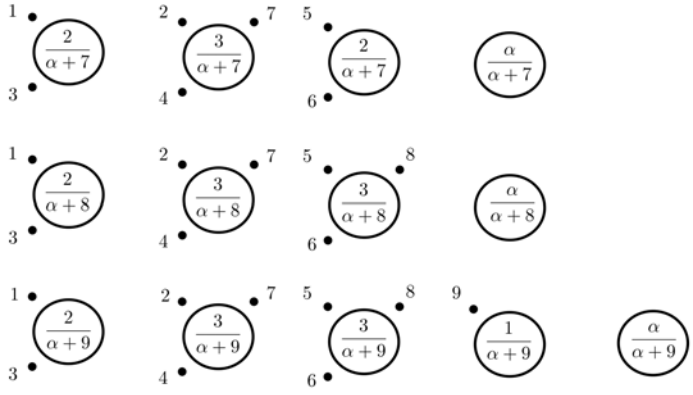


Figure 3.3: Illustration of the Chinese restaurant process. Circles correspond to tables in the restaurant, while numbers correspond to customers sitting on tables. The first row represents a configuration after 6 customers have arrived at the restaurant defining a partition with 3 clusters $\{\{1,3\},\{2,4,7\},\{5,6\}\}$. In the final partition (bottom row) 9 customers have been allocated to 4 clusters $\{\{1,3\},\{2,4,7\},\{5,6,8\},\{9\}\}$.

number of tables is:

$$\mathbb{E}\{K|N\} = \sum_{i=1}^{N} \frac{\alpha}{\alpha + i - 1}. \tag{3.35}$$

This quantity converges asymptotically to $O(\alpha \log N)$, so that the number of clusters grows logarithmically with the number of customers.

3.3.4 Infinite Mixture Models

We can use a DP to model different types of data by covolving the random measure $G \sim DP(\alpha, G_0)$ with different kernels. This is the idea behind DPMMs, was firstly proposed by Antoniak (1974). A DPMM is defined by the following generative process:

$$G \sim DP(\alpha, G_0), \tag{3.36}$$

$$\theta_i \sim G, \quad \forall i, \tag{3.37}$$

$$\boldsymbol{x}_i \sim P(\theta_i), \quad \forall i, \tag{3.38}$$

where $\{\theta_i\}_{i=1:\infty}$ is a set of latent finite-dimensional random variables and $\boldsymbol{x}_i$ is sampled from a parametric distribution $P(\theta_i)$ parametrized by θ_i.

Introducing the parametric distribution $P(\cdot)$ is essential when dealing with continuous data. Remember that the realizations of a DP are discrete with probability one, so multiple observations among the set $\{\boldsymbol{x}_i\}_{i=1:N}$ take the exact same value. In the DPMM this constraint is eliminated. Due to the discrete nature of G different elements of $\{\theta_i\}_{i=1:\infty}$ can take the same value, but by choosing $P(\theta_i)$ non-atomic, the elements of the set $\{\boldsymbol{x}_i\}_{i=1:\infty}$ are different with probability one. In this sense, this generative model can be seen as a MM in which the values $\boldsymbol{x}_i$ that correspond to the same value of θ_i are allocated to the same cluster.

If we denote by $\{\boldsymbol{\phi}_k\}_{k=1:\infty}$ the set of unique values in $\{\theta_i\}_{i=1:\infty}$, we can rewrite the model making use of the SB representation:

$$\boldsymbol{\phi}_k \sim G_0, \quad \forall k, \tag{3.39}$$

$$\pi_k = v_k \prod_{\ell=1}^{k-1}(1-v_\ell), \quad v_k \sim \text{Beta}(1,\alpha), \quad \forall k, \tag{3.40}$$

$$x_i \sim \sum_{k=1}^{\infty} \pi_k P(\boldsymbol{\phi}_k), \quad \forall i, \tag{3.41}$$

where we can see the correspondence with the FMM defined by Equations 3.3, 3.4 and 3.5. Effectively, the parametric likelihood $P(\cdot)$ imposes a notion of distance in the space to which the observations belong, $\mathcal{X}$, while the DP prior allows a global nonparametric distribution whose complexity depends on the number of data points.

In the same way we did in the case of FMM, we can introduce a set of auxiliary latent random variables to explicitly model the allocations of the data to the different clusters:

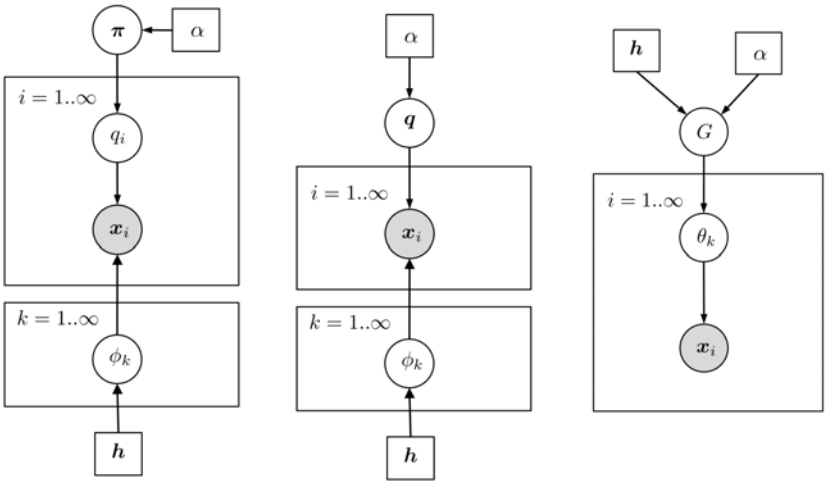
$$\phi_k \sim G_0, \quad \forall k, \tag{3.42}$$

$$\pi_k = v_k \prod_{\ell=1}^{k-1}(1-v_\ell), \quad v_k \sim \text{Beta}(1,\alpha), \quad \forall k, \tag{3.43}$$

$$q_i|\boldsymbol{\pi} \sim \text{Mult}(\boldsymbol{\pi}), \quad \forall i, \tag{3.44}$$

$$\boldsymbol{x}_i|\boldsymbol{\phi} \sim P(\boldsymbol{\phi}_{q_i}), \quad \forall i. \tag{3.45}$$

If we integrate out the proportions $\boldsymbol{\pi}$ and the parameters of the components $\boldsymbol{\phi}$,



(a) Stick-breaking based representation

(b) Chinese Restaurant based representation

(c) Dirichlet Process based representation

Figure 3.4: Graphical equivalent representations for a Dirichlet Process Mixture Model

we get another useful representation in terms of the CRP prior:

$$\boldsymbol{q}|\alpha \sim \text{CRP}(\alpha), \tag{3.46}$$

$$\mathcal{D}|\boldsymbol{q}, \boldsymbol{\alpha}, \boldsymbol{h} \sim \int_{\Phi} \prod_{k=1}^{\infty} \left[\left(\prod_{i:q_i=k} P(\boldsymbol{x}_i|\boldsymbol{\phi}_k) \right) dP(\phi_k|\boldsymbol{h}) \right], \tag{3.47}$$

which is equivalent to the FMM described by Equations 3.8 and 3.10. In Figure 3.4 we can see the different representations of a DPMM.

As in other BNP models, we assume an infinite number of parameters a priori, but we end up with a finite number of parameters a posteriori after observing a set $\mathcal{D} = \{x_i\}_{i=1:N}$. To compute the posterior of $\boldsymbol{q}$ given a collection of data $\mathcal{D}$, we need to sum over all possible partions of the data defined by $\boldsymbol{q}$. The number of terms of this sum grows even faster that the number of terms in a FMM because now we are considering partitions with different number of clusters. Specifically, the number of terms is given by the Bell number growing super-exponentially with the cardinality of the data $\mathcal{D}$ (see Figure 3.5). In Section 3.3.5 we review the main inference algorithms to approximate the posterior of the parameters in a DPMM.

Finally, in addition to considering a family of models of variable complexity,

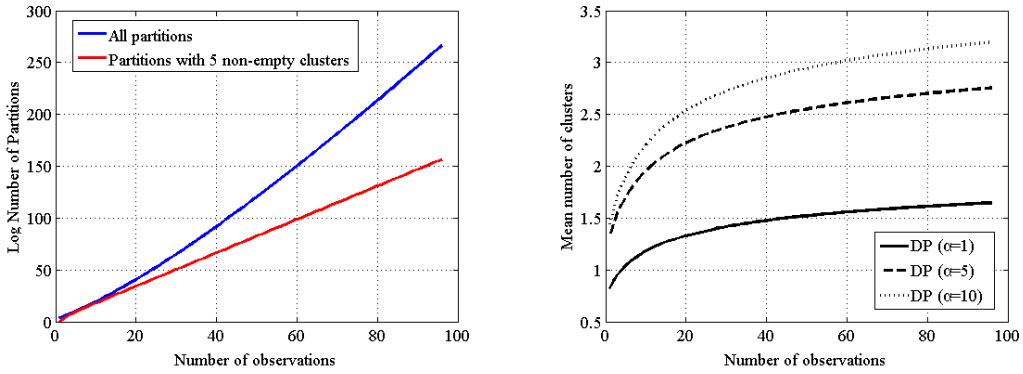


Figure 3.5: (Left) In red, evolution of the total number of possible partitions of N data in a Dirichlet Process Mixture Model (given by the Bell number). In blue, evolution of the total number of possible partitions of N data in a Finite Mixture Model with K components (given by the Stirling's number of the second kind). (Right) Evolution of the mean number of clusters of a Dirichlet Process Mixture model for different values of α

another advantage of DPMM becomes clear when we compute the predictive distribution:

$$P(\boldsymbol{x}_{N+1}|\{\boldsymbol{x}_i\}_{i=1:N}) = \sum_{\{q_i\}_{i=1:N+1}} \int_{\boldsymbol{\phi}} P(\boldsymbol{x}_{N+1}|q_{N+1},\boldsymbol{\phi})P(q_{N+1}|\{q_i\}_{i=1:N},\alpha)dP(\{q_i\}_{i=1:N},\boldsymbol{\phi}|\{\boldsymbol{x}_i\}_{i=1:N},\alpha). \tag{3.48}$$

We can see that the CRP appears in the predictive distribution and, therefore, a DPMM allows new data to exhibit a previously unseen cluster.

3.3.5 Inference

Despite its early discovery by Ferguson (1973), the DP remained unnoticed by the machine learning community until recently. In the last years, the proliferation and wide accessibility of powerful computing resources, together with the development of efficient approximate inference algorithms, has awoken the interest in DP as a valuable statistical modeling tool, specially as a flexible prior for MMs. In this section, we give an overview on some of the most widely-used inference algorithms that approximate the posterior. A thorough description of inference methods in

BNP models is beyond the scope of this chapter, but details on the inference algorithms for the models developed in this thesis are given in Chapters 4 and 5.

The most common inference problem in DPMMs is, given a set of observed data $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1:N}$, compute the posterior distribution of the auxiliary latent allocation variables $\boldsymbol{q}$ and the parameters of the components $\boldsymbol{\phi}$, i.e. $P(\boldsymbol{q}, \boldsymbol{\phi}|\mathcal{D})$. To do so, we can resort to two different families of approximate inference algorithms: Monte Carlo (MC) and Variational methods.

MC methods (Metropolis and Ulam, 1949), propose to approximate the posterior distribution $P(\boldsymbol{q}, \boldsymbol{\phi}|\mathcal{D})$ by a set of samples. Among MC methods, the most widely used subfamily of inference methods for approximating the posterior distribution in DPMM is the family of Markov Chain Monte Carlo (MCMC) methods (Andrieu et al., 2003). MCMC methods define a Markov Chain on the latent variables such that its stationary distribution is precisely $P(\boldsymbol{q}, \boldsymbol{\phi}|\mathcal{D})$, so by drawing samples from this chain, we eventually get samples from the target distribution, i.e. $P(\boldsymbol{q}, \boldsymbol{\phi}|\mathcal{D})$.

The first MCMC algorithms for DPMM where the so-called *Marginal Samplers*. These sampling algorithms rely on marginalizing the random measure $G \sim DP(\alpha, G_0)$ removing the problem of dealing with an infinite-dimensional object. Therefore, these methods use a representation of the DPMM based on the predictive distribution, i.e. *Polya-Urn* Scheme/CRP Process. Some of the earliest marginal samplers for DPMM were proposed by Escobar (1994), Escobar and West (1995), MacEachern (1994) and MacEachern and Müller (1998). An overview of marginal samplers for DPMMs with conjugate priors was published by Neal (2000). In particular, the author analyze the performance of different marginal samplers based on the Gibbs algorithm, i.e. the Markov chain is built by considering the conditional distribution of each latent variable given the rest, and a collapsed version of it where the parameters of the components are integrated out. In addition, the authors review the existing algorithms for DPMM with non-conjugate priors and published a new algorithm, in which the probability of an observation being allocated to a new cluster is approximated by a set of auxiliary

clusters.

The main limitation of Gibbs sampling is that it is based on incremental proposals. As an alternative, non-incremental MCMC methods based on split-merge moves have been proposed by Green and Richardson (2001) and Jain and Neal (2004). The method by Green and Richardson (2001) is based in a Reversible Jump Markov Chain Monte Carlo (RJMCMC) procedure while Jain and Neal (2004) rely on a Metropolis Hasting (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) with split-merge proposals (an extension to non-conjugate models was later published by Jain and Neal (2007)).

Finally, another class of marginal samplers was proposed by Liu (1996). Like other marginal samplers, it is based on the *Polya-Urn*/CRP representation of a DP. However, instead of using MCMC, it relies on a Sequential Monte Carlo (SMC) strategy. The advantage is that it does not rely on an underlying Markov chain. Instead, i.i.d. samples are drawn to create an importance sample. The disadvantage is that the importance weights can have high variance giving raise to inaccurate estimators. Several publications address this problem (MacEachern et al., 1999; Fearnhead, 2004; Ülker et al., 2010; Ulker et al., 2011).

A second family of MCMC inference methods, called *Conditional Samplers*, is based on the SB representation of the DP. Instead of integrating out the infinite dimensional random measure $G \sim DP(\alpha, G_0)$, *Conditional Samplers* focus on sampling from a finite but large enough number of atoms of G. One of the earliest *Conditional Samplers* was devised by Ishwaran and James (2001), who proposed a deterministic truncation level and bound the error committed by using this finite-dimensional approximation. Several methods have been proposed to sample the posterior without resorting to a finite-dimensional approximation. In this line of work, Papaspiliopoulos and Roberts (2008) propose to use a retrospective sampling scheme to draw samples from a SB process using a uniform auxiliary variable, while Walker (2007) directly augments the generative model and applies slice sampling (Neal, 2003). With the same goal of avoiding a deterministic truncation level several publications can be found in the literature (Muliere and Tardella, 1998;

Papaspiliopoulos, 2008; Kalli et al., 2011).

An alternative to MC methods is Variational inference (Jordan et al., 1999; Ghahramani and Beal, 2001; Wainwright and Jordan, 2008). This encompasses a wide range of deterministic approximate inference algorithms in which the basic idea is to approximate our complex target distribution, i.e. the posterior distribution of the parameters $P(\boldsymbol{q}, \boldsymbol{\phi}|\mathcal{D})$, by a variational distribution $Q(\boldsymbol{q}, \boldsymbol{\phi})$ that is constrained to belong to a tractable family $Q(\boldsymbol{q}, \boldsymbol{\phi}) \in \mathcal{Q}$. Therefore, these methods turn an inference problem into an optimization problem. Considering different families $\mathcal{Q}$ leads to different approximations in which there is always a trade-off between accuracy and simplicity.

The advantages of variational inference methods is that they are generally faster and, unlike MCMC, it is easy to check the convergence. The main disadvantages are that they do not converge to the true posterior distribution and they can be applied to a less broad class of models.

Regarding the DPMM, variational inference was originally applied using a SB representations of the DP. In the seminal paper published by Blei and Jordan (2006), the authors proposed to use a *Mean-Field* approach, in which the joint variational distribution over the latent variables is fully factorized. They truncate the variational distribution to a finite number of components, but they do not need to truncate the posterior $P(\boldsymbol{q}, \boldsymbol{\phi}|\mathcal{D})$. Kurihara et al. (2006) proposed an extension in which they use an infinite dimensional variational distribution by fixing it to the prior after a certain truncation level. In addition they propose to use kd-trees to speed up the inference. Kurihara et al. (2007) propose to integrate out the weights of the components of the DPMM following the philosophy of *Marginal Samplers*. They also study the effects of label reordering in the SB representation. Finally, Zobay (2009) performs a systematic study of the previous mean-field methods.

Beyond *Mean-Field* approximations, Fan and Bouguila (2013) use Expectation Propagation (EP) (Minka, 2001) based on the SB representation. Finally, Wang and Blei (2012) proposed to use a locally collapsed variational inference algorithm, which enables truncation-free variational inference. They rely again on the SB

	Monte Carlo Inference	Variational Inference
Marginal representation	Escobar (1994), Escobar and West (1995), MacEachern (1994), MacEachern (1994), Neal (2000), Green and Richardson (2001), Jain and Neal (2004), Jain and Neal (2007)), Liu (1996), MacEachern et al. (1999), Fearnhead (2004), Ülker et al. (2010), Ulker et al. (2011)	Kurihara et al. (2007), Wang and Blei (2012)
Conditional representation	Ishwaran and James (2001), Papaspiliopoulos and Roberts (2008), Walker (2007) , Muliere and Tardella (1998), Papaspiliopoulos (2008), Kalli et al. (2011)	Blei and Jordan (2006), Kurihara et al. (2006), Fan and Bouguila (2013), Wang and Blei (2012)

Table 3.1: Main approximate inference algorithms for Dirichlet Process Mixture Models. The marginal representation is based on the Pólya Urn/Chinese Restaurant process while the Conditional representation is based on the Stick-breaking construction.

representation, but they use as a subroutine a collapsed Gibbs sampler based on the CRP representation. In Table 3.1, a summary of the main approaches to perform approximate inference in DPMMs is presented.

3.4 Other Bayesian Nonparametric Priors

In this chapter, we have focused on BNP priors for MMs given its relevance for the rest of the thesis. However, this is not the only application of BNP models. An extension called the Hierarchical Dirichlet process (HDP) (Teh et al., 2003) is useful as a prior for HMMs with a potentially unbounded number of hidden states. It is also useful to build a nonparametric version of Latent Dirichlet Allocation model (Blei et al., 2003), in which each document corresponds to a group, and the number of topics is potentially infinite. The Indian Buffet Process (IBP) (Griffiths and Ghahramani, 2005) is a BNP prior that allows to build FMs in which the number of features is not known a priori. The Gaussian Process (GP) (Rasmussen and Williams, 2006) is another BNP prior generally used as a discriminative approach to non-linear regression and classifications problems. These are only a few examples. In general, the choice of an appropriate stochastic process depends on the problem at hand. For more information about the available range of BNP priors available in the literature, we refer the interested reader to Hjort et al. (2010); Gershman and Blei (2012); Orbanz and Teh (2010) and the references therein.

4

Identifying Communities of Annotators

4.1 Introduction

One of the key reasons why crowdsourcing has become such a profitable alternative compared to traditional markets is its flexibility (see Chapter 2). Generally, each annotator labels only a small subset of the instances, and each instance is labeled by a small subset of the annotators that are registered in the system. In this way, we take advantage of the fact that workers are more willing to accept shorter tasks for a lower payment rate than the one they would accept for longer tasks (Buhrmester et al., 2011; Hoßfeld et al., 2014), i.e. a annotator may accept to label 100 photos at 0.1 euros but may not take a task of looking at 10000 unless we increase the reward per photo.

As a consequence, in many crowdsourcing problems we deal with a highly sparse matrix of annotations provided by a pool of annotators whose expertise

is unknown. This lack of information can seriously harm the performance of the system, in particular when the platform relies on complex models governed by lot of parameters. In addition, the amount of available information varies with time. The level of sparsity will be specially high at the beginning of the project when the number of annotators is scarce, being the harming more severe.

In the literature, this problem is commonly known as the Cold Start Problem (CSP). The CSP is not a specific problem of crowdsourcing. Actually, the problem is well-known in the Collaborative Filtering (CF) literature (Shi et al., 2014; Su and Khoshgoftaar, 2009), where it happens whenever a new annotator or item enters the system and the lack of information makes it difficult to find similar ones. Different strategies have been proposed to alleviate it, but we can classify them in two big groups:

- The ones that use external information about the annotators (personal information, social networks, etc.) and the items (taxonomies, price, etc.) to improve the similarity metric, e.g. (Z. and G., 2009; George and Merugu, 2005; Koren et al., 2009; Luo et al., 2012; Z. et al., 2005)

- The ones that rely on reducing the number of parameters of the model, e.g. (Leung et al., 2008; Loh et al., 2009; Weng et al., 2008; Heung-Nam et al., 2010)

We focus on the second alternative because for the first one we need external information that is not always available.

Another problem that has recently received significant attention in crowdsourcing is the detection of groupings among the labelers (Simpson et al., 2011, 2013). In most crowdsourcing applications we can identify several types of annotators: experts, novices, spammers and even malicious or adversarial annotators. Identifying these groups of annotators and learning about their properties is useful to design efficient crowdsourcing strategies that minimize the overall cost, selecting the most suitable annotators for each labeling task.

Usually, the detection of grouping of labelers is tackled in a post processing

step, after the ground truth has been estimated from their annotations (see Section 4.2.4). This approach has several problems. To begin with, as in any cascade model, the errors in the first stage (identifying the underlying ground truth of the instances) propagates to the second stage (identifying the different communities of annotators that are present in our pool). Secondly, the estimation of the ground truth is done without considering the clustering structure of the annotators, and so no the first stage does not get any benefit from correctly solving the second one.

In this chapter, we propose two unsupervised transductive Bayesian Nonparametric (BNP) models to combine the labels provided by the annotators in a crowdsourcing scenario, taking into account the presence of clusters of annotators. Our models jointly solve the problem of the estimation of ground truth and the problem of identification of clusters of annotators and their properties. The estimation of the ground truth improves the clustering of the annotators and vice versa, thus performing better than current state-of-the-art (Kim and Ghahramani, 2012; Simpson et al., 2011). The overall improvement in both tasks is particularly important in the early stages of a crowdsourcing project, when the CSP is more severe. In this case, algorithms that estimate the properties of each annotator independently, without considering the dependencies among them, tend to provide poor estimates due to the large number of parameters to infer, and may perform even worse than majority voting (see Section 4.3). Our model is able to adapts its complexity depending on the amount of information we have, i.e. the more information we have the more complex models we can consider.

In the first model, we propose a clustering structure using a Chinese Restaurant Process (CRP) prior (Pitman, 2002). In this model, all the annotators that belong to the same cluster share the same parameters governing the way they label instances, and therefore, they have the same behavior. This improves the performance, specially when the input matrix of annotations is sparse. However, forcing all the annotators to share the same exact parameters, is a strong assumption that might lead to groupings with a large number of cluster. Therefore, these groupings are difficult to interpret and not very useful to extract meaningful in-

formation about the annotators present in our corwdsourcing platform. To relax this assumption we propose a second model in which annotators that belong to the same cluster are modeled as having similar parameters, but allows each annotator to have its own parameters using a hierarchical Bayesian approach.

In this chapter, we rely on a BNP model, because we are not only interesting in having an accurate model but also in having an interpretable one. In the experiments (Section 4.3) we show that the error rates between the two proposed models are not significantly different. However, the second one is interpretable, in the sense that it identifies each kind of clusters and it reports the least number of them. The interpretability of the model is principal to us, because we want to use the model to identify the 'good' annotators and be able to reward them accordingly, while other models are not able to provide this information.

The rest of the chapter is organized as follows. In Section 4.2, we present the two new generative models for crowdsourcing that take into account the clustering structure of the annotators. In Section 4.2.3, we propose efficient Markov Chain Monte Carlo (MCMC) inference algorithms for estimating the different groups of annotators as well as the ground truth. In Section 4.2.4, we review related literature on crowdsourcing and the identification of annotator clusters in the context of crowdsourcing. In Section 4.3, we validate our model on synthetic data and we perform several experiments on real datasets to show the advantages of our models over state-of-the-art algorithms. Finally, we conclude this chapter in Section 4.4 and present some possible extensions for the future.

4.2 Hierarchical Bayesian Combination of Classifiers

In this section, we propose two different models that capture the dependencies among the annotators by assuming an underlying clustering structure.

Both algorithms receive as input a set of noisy labels $\boldsymbol{Y} \in \{1, ..., C\}^{N \times L}$ provided by L annotators for N instances. The element $y_{i\ell}$ represents the label given by the annotator ℓ to the instance i and it is 0 if the annotator did not label the corresponding instance. Notice that this matrix $\boldsymbol{Y}$ is highly sparse in the early

stages of a crowdsourcing application due to the CSP.

The output of the algorithms is the set of true but unknown labels of the instances $\boldsymbol{z} \in \{1, ..., C\}^N$, where z_i indicates the true label estimate of the instance i. In theory, nothing prevent us from using different support for the true label and the annotations, i. e. $z_i \in \{1, ..., T\}$ and $y_{i\ell} \in \{1, ..., C\}$. For example, Simpson et al. (2011) use a dataset that contains scores given by individual volunteer citizen scientists to candidate supernova images. They answer a set of linked questions to classify the image in "very unlikely● a supernova","possibly a supernova" and "very likely a supernova". However, each image is either "supernova" or "not supernova". In this case $C = 3$ and $T = 2$. However, in this chapter we focus only in the case $C = T$, given that is general enough and it simplifies the notation.

We denote by $[L] = \{1, 2, ..., L\}$ the set of indexes of the annotators and by $\boldsymbol{\pi}_L$ a partition of $[L]$. A partition is a collection of mutually exclusive, mutually exhaustive and non-empty subsets called clusters. Some examples of partitions for $L = 5$ would be $\{\{1, 2, 3\}\{4, 5\}\}$, $\{1, 2, 3, 4, 5\}$ or $\{\{1\}\{2, 5\}\{3, 4\}\}$. We denote the cluster assignment of the annotator ℓ with a variable q_ℓ such that $q_\ell = m$ denotes the event that the annotator ℓ is assigned to cluster $m \in \boldsymbol{\pi}_L$. Therefore, the set of variables $\boldsymbol{q} = \{q_\ell\}_{\ell \in [L]}$ defines a partition of the annotators.

Finally, it is important to notice that $\boldsymbol{Y}$ is the only observed variable in the models, being fully unsupervised. However, if the ground truth of some instances is know, the model can easily incorporate that piece of information to provide a better estimation of the unobserved parameters.

4.2.1 Clustering based Bayesian Combination of Classifiers

Firstly, we propose a model for annotators in which they can belong to different clusters. In each cluster all the annotators have the same properties. We name it Clustering based Bayesian Combination of Classifiers (cBCC) (see Figure 4.2)

and it has the following observation model:

$$y_{i\ell}|z_i, \boldsymbol{\pi}, \boldsymbol{\Psi} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\Psi}_{z_i}^{q_\ell})$$
$$z_i|\boldsymbol{\tau} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\tau}).$$

We assume that all the annotators that belong to cluster $m \in \boldsymbol{\pi}$ share the same properties, i.e. the same confusion matrix $\boldsymbol{\Psi}^m \in [0,1]^{C\times C}$, where Ψ_{tc}^m is the probability that a annotator allocated in cluster m labels an instance as $y = c$ when the ground truth is $z = t$. Therefore, Ψ_{tc}^m is a right stochastic matrix as $\Psi_{tc}^m \geq 0$ and $\sum_c \Psi_{tc}^m = 1$. In addition, we use the notation $\boldsymbol{\Psi}_t^m \in \mathcal{S}^C$ to denote the row t of $\boldsymbol{\Psi}^m$, where $\mathcal{S}^C$ is the C-dimensional probability simplex. The annotation provided by an annotator belonging to cluster $m \in \boldsymbol{\pi}$ for the instance i is sampled from a discrete distribution whose parameters are given by the row of $\boldsymbol{\Psi}^m$ indicated by the value of its ground truth. The ground truth is sampled from a discrete distribution with parameters $\boldsymbol{\tau}$, where the component τ_t of $\boldsymbol{\tau} \in \mathcal{S}^C$ is the probability of the ground truth z being equal to $t \in \{1, ..., C\}$.

We also need to define priors to complete the Bayesian model. In particular, we choose to use conjugate priors:

$$\boldsymbol{\Psi}_t^m|\boldsymbol{\beta}, \boldsymbol{\eta} \sim \text{Dir}(\beta_t \boldsymbol{\eta_t})$$
$$\boldsymbol{\tau}|\epsilon, \boldsymbol{\mu} \sim \text{Dir}(\epsilon\boldsymbol{\mu}),$$

where we use a Dirichlet prior on each of the rows of the confusion matrices in which $\boldsymbol{\eta}_t \in \mathcal{S}^C$ is the mean value of $\boldsymbol{\Psi}_t^m$ while $\beta_t \in \mathbb{R}_+$ is related to its precision, i.e. the bigger it is this value, the more concentrated is the dirichlet distribution around the mean parameter. Notice that this is an over parametrization of the Dirichlet distribution, which only needs C parameters to be fully determined. However, this decomposition is useful to interpret the results as well as for the development of the inference algorithms in Section 4.2.3. Likewise, we set a Dirichlet prior on $\boldsymbol{\tau}$, where $\boldsymbol{\mu} \in \mathcal{S}^C$ is the mean and $\epsilon \in \mathbb{R}_+$ relates to the precision.

We could use a parametric model in which the cardinality of the partition $M = |\boldsymbol{\pi}|$ is fixed a priori. Unfortunately, in this case the inferences are sensitive

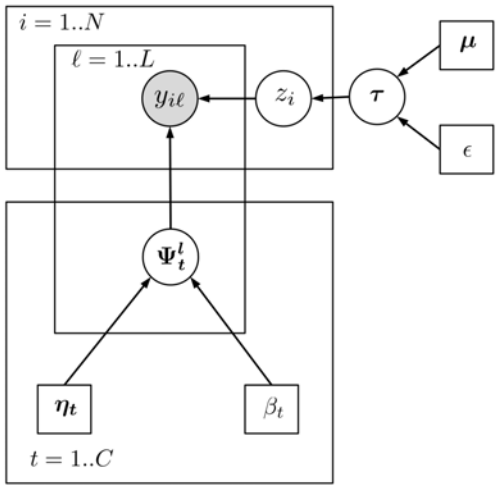


Figure 4.1: Graphical model representation of the Independent Bayesian Combination of Classifiers model.

to the value of M chosen. In the limiting case $M = 1$ the model is equivalent to majority voting and does not capture the differences in the behavior of the different annotators. If M is too large the model does not take advantage of the presence of clusters of annotators. In the limiting case when $M = L$ each annotator becomes a singleton cluster, and the model does not capture the dependencies among the annotators.

To find M we could use traditional model selection strategies like cross-validation (Stone, 1974) or Bayesian Information Criterion (Fraly and Raftery, 1998). This approach has two limitations. The first one is that in many situations we do not have access to a validation set for which $\boldsymbol{z}$ is known. The second one, is the high computational complexity. These methods generally involve running the algorithm for every possible candidate model, i.e. every possible M, and then keeping the one that maximize a particular metric.

An alternative pathway is to set a prior on the space of all possible partitions following a BNP approach. We denote by $\mathcal{P}_L$ the space of all partitions over the annotators space $\pi_L \in \mathcal{P}_L$ and we set a prior on an infinite number of annotators, i.e. on partition $\boldsymbol{\pi} \in \mathcal{P}_\infty$. A (exchangeable and consistent) prior on $\mathcal{P}_\infty$ is the

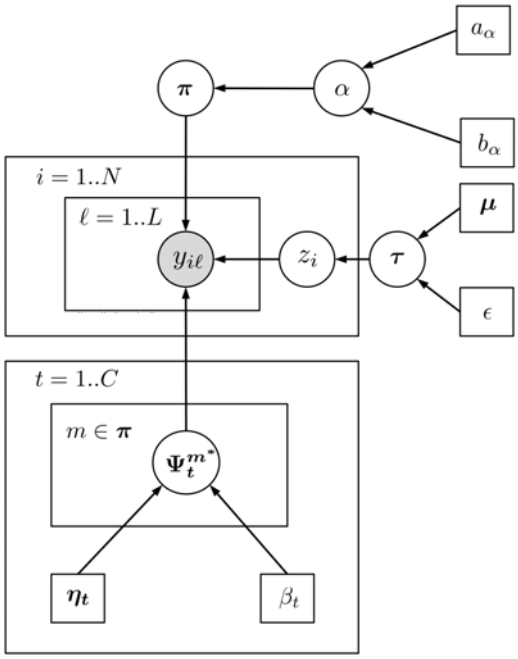


Figure 4.2: Graphical model representation of the Clustering based Bayesian Combination of Classifiers model.

CRP (see Chapter 3):

$$\boldsymbol{\pi}|\alpha \sim \text{CRP}(\alpha). \tag{4.1}$$

We can generate samples from this prior using the following conditional distributions:

$$p(q_\ell = m|\boldsymbol{\pi}^{\neg\ell}, \alpha) \propto \begin{cases} |m|^{\neg\ell}, & m \in \boldsymbol{\pi}^{\neg\ell} \\ \alpha, & m = \emptyset, \end{cases}$$

where $|m|$ represents the number of annotators in cluster m and $|m|^{\neg\ell}$ is equal to $|m|$ excluding annotator ℓ. We denote by $\boldsymbol{\pi}^{\neg\ell}$ the partition with the annotator ℓ removed and $q_\ell = \emptyset$ denotes the event that annotator ℓ is assigned to a new cluster. α is the so called concentration parameter and control the a priori probability of generating new clusters. We further place a gamma prior over the concentration parameter α:

$$\alpha|a_\alpha, b_\alpha \sim \text{Gamma}(a_\alpha, b_\alpha). \tag{4.2}$$

If α tends to infinity, every annotator is allocated to a singleton cluster. If α tends

to 0, all the annotators share the same confusion matrix and the model produces a majority voting solution.

In general, the CRP assigns more mass to partitions with a small number of clusters. This is sensible in our case, because when the number of annotations is scarce, majority voting may perform better than more elaborate algorithms since there is no enough information to estimate the individual properties of the annotators. In this case, the CRP prior dominates and therefore all annotators are allocated to the same cluster. When the number of annotations increase, the likelihood term dominates and more complex models are taking into consideration, i.e. partitions with a higher number of clusters.

Finally, this clustering structure induces a correlation between the different annotators. We can compute the correlation structure that it is introduced among the annotators as a consequence of this clustering. The correlation a priori among two annotators ℓ and ℓ' is:

$$\begin{aligned}&\text{Corr}(\mathbb{I}(y_{i\ell}=a),\mathbb{I}(y_{i\ell'}=b)|z_i=t)=\\&\begin{cases}-\left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\sqrt{\frac{\eta_{ta}\eta_{tb}}{(1-\eta_{ta})(1-\eta_{tb})}} & a\neq b\\ \left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right) & a=b.\end{cases}\end{aligned} \tag{4.3}$$

Here, $\mathbb{I}(.)$ represents the indicator function. The proof is provide in the Appendix A. In Section 4.2.4, we show how this model relates to other state-of-the-art algorithms.

4.2.2 Hierarchical Clustering based Bayesian Combination of Classifiers

In the cBCC model, the annotators that belong to the same cluster share the same confusion matrix. This is a nice way of reducing the number of parameters of the model, and therefore, it helps to alleviate the effect of the CSP when the input matrix is highly sparse. However, in a practical situation, each annotator has a behavior that is somehow different from every other annotator, but it is in some sense similar to the behavior of annotators that are allocated to its cluster. To

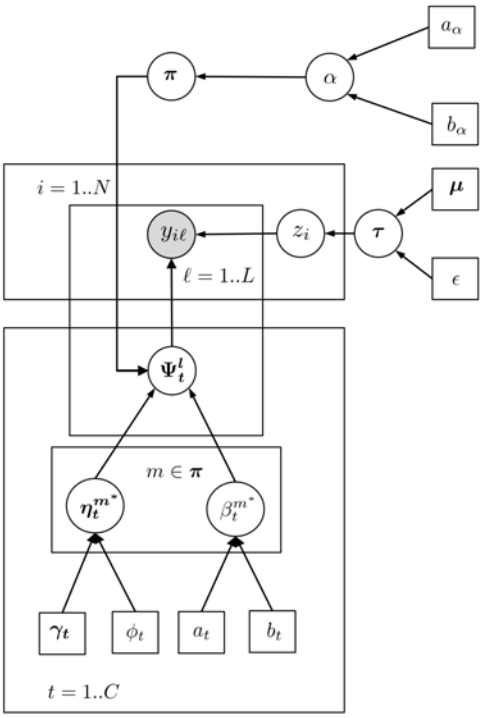


Figure 4.3: Graphical model representation of the Hierarchical Clustering based Bayesian Combination of Classifiers model

overcome this restriction we propose a hierarchical extension of the cBCC model called Hierarchical Clustering based Bayesian Combination of Classifiers (hcBCC) depicted in Figure 4.3. The observation model is the following:

$$y_{i\ell}|z_i, \boldsymbol{\Psi} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\Psi}^{\ell}_{z_i})$$
$$z_i|\boldsymbol{\tau} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\tau}).$$

Now each annotator has its own confusion matrix $\boldsymbol{\Psi}^{\ell}$ in contrast to the cBCC model where we had a confusion matrix per cluster $\boldsymbol{\Psi}^{m}$. To capture the similarity between annotators that belong to the same cluster we use the following hierarchical prior:

$$\boldsymbol{\Psi}^{\ell}_t|\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\eta} \sim \text{Dir}(\beta^{q_\ell}_t \boldsymbol{\eta}^{q_\ell}_t)$$
$$\beta^{m}_t|a_t, b_t \sim \text{Gamma}(a_t, b_t)$$
$$\boldsymbol{\eta}^{m}_t|\boldsymbol{\phi}, \boldsymbol{\gamma} \sim \text{Dir}(\phi_t \boldsymbol{\gamma}_t)$$
$$\boldsymbol{\tau}|\epsilon, \boldsymbol{\mu} \sim \text{Dir}(\epsilon \boldsymbol{\mu}).$$

In this way, the confusion matrices of all annotators that belong to the same cluster m are generated from the same distribution. In particular, each of the rows of the confusion matrices of all the annotators that belong to cluster m, i.e. $\{\boldsymbol{\Psi}^{\ell}_t : q_\ell = m\}$, are i.i.d. samples from the same Dirichlet distribution whose parameters are β^{m}_t and $\boldsymbol{\eta}^{m}_t$. A Dirichlet prior is set on the vector $\boldsymbol{\eta}^{m}_t$ while a gamma prior is set on the scalar β^{m}_t. Finally, for $\boldsymbol{\pi}$ and α we, respectively, use the same priors given by Equations 4.1 and 4.2. With this we have a model where we no longer cluster the confusion matrices of the annotators, but the distributions that generate them.

Notice that the vector $\boldsymbol{\beta}^{m}$ governs the variability among the annotators that belong to the same cluster m. The bigger these values are, the lower is the intra-cluster variability, i.e. the probability mass of the corresponding Dirichlet distribution gets more concentrated around its mass. If we make each of the components of $\boldsymbol{\beta^m}$ tend to infinity, then the variability among the annotators tend to 0 and the model becomes equivalent to the cBCC model forcing all the annotators of the

cluster to share the exact same confusion matrix. In this way, this model can be seen as a generalization of some state-of-the-art methods (see Section 4.2.4).

4.2.3 Inference

Computing the posterior distribution of the clusters allocation, the properties of the annotators and the estimated ground is intractable, so we resort to MCMC inference. In this section we propose to use a collapsed Gibbs sampling together with the corresponding auxiliary variables whenever it is not possible to compute the conditional distributions due to non-conjugacies.

4.2.3.1 cBBC

We use a collapsed Gibbs sampling algorithm where we integrate out the variables $\boldsymbol{\Psi}^m$ and $\boldsymbol{\tau}$. Collapsed Gibbs sampling has been proven to be more efficient than its uncollapsed counterpart since it is sampling in a lower dimensional space (Liu et al., 1994) . In our case, this integration can be done analytically due to the use of conjugate priors, obtaining the following new set of equations:

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{\eta}, \boldsymbol{\beta}) &= \prod_m \prod_t \left[\frac{\Gamma(\beta_t)}{\Gamma(n_{mt} + \beta_t)} \prod_c \frac{\Gamma(n_{mtc} + \beta_t \eta_{tc})}{\Gamma(\beta_t \eta_{tc})} \right], \\ p(\boldsymbol{z}|\epsilon, \boldsymbol{\mu}) &= \frac{\Gamma(\epsilon)}{\Gamma(N + \epsilon)} \prod_t \frac{\Gamma(n_m + \epsilon \mu_t)}{\Gamma(\epsilon \mu_t)}, \end{aligned} \tag{4.4}$$

where $\Gamma(\cdot)$ denotes the gamma function. We denote $n_{i\ell mtc} = \mathbb{I}(z_i = t, y_{i\ell} = c, y_{i\ell} \neq 0, q_\ell = m)$, and when an index of this variable is omitted we assume it is summed out. For example, n_{mtc} represents the number of annotations equal to c provided by the annotators of cluster m for the set of instances whose ground truth is equal to t. Then, we use Gibbs sampling to infer the value of the ground truth $\boldsymbol{z}$, the clusters of annotators $\boldsymbol{\pi}$, as well as the hyper parameters of the CRP, conditioned on the observed variables $\boldsymbol{Y}$.

Firstly, to update the cluster assignment of annotator ℓ, we need the conditional

distribution of q_ℓ given the rest of the variables:

$$p(q_\ell = m|\text{rest}) \propto \begin{cases} n_m^{\neg\ell} \times \prod_t \frac{\Gamma(n_{mt}^{\neg\ell} + \beta_t)}{\Gamma(n_{mt}^{\neg\ell} + n_{\ell t} + \beta_t)} \prod_t \prod_c \frac{\Gamma(n_{mtc}^{\neg\ell} + n_{\ell tc} + \beta_t \eta_{tc})}{\Gamma(n_{mtc}^{\neg\ell} + \beta_t \eta_{tc})}, & m \in \boldsymbol{\pi}^{\neg\ell} \\ \alpha \times \prod_t \frac{\Gamma(\beta_t)}{\Gamma(n_{\ell t} + \beta_t)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc} + \beta_t \eta_{tc})}{\Gamma(\beta_t \eta_{tc})}, & m = \emptyset \end{cases}$$

where $q_\ell = \emptyset$ denotes the event that annotator ℓ is assigned to a new cluster. The quantities $n_{mt}^{\neg\ell}$ and $n_{mtc}^{\neg\ell}$ are defined in the same way as n_{mt} and n_{mtc} respectively, but excluding the annotator ℓ. The complexity of updating the $\boldsymbol{q}$ variables is $O(LMTC)$. To sample the estimate of the ground truth z_i of each instance conditioned on the rest of the variables, the required conditional distribution is:

$$p(z_i = t|\text{rest}) \propto \left(n_t^{\neg i} + \epsilon\mu_t\right) \times \prod_m \left[\frac{\Gamma(n_{mt}^{\neg i} + \beta_t)}{\Gamma(n_{mt}^{\neg i} + n_{im} + \beta_t)} \prod_c \frac{\Gamma(n_{mtc}^{\neg i} + n_{imc} + \beta_t \eta_{tc})}{\Gamma(n_{mtc}^{\neg i} + \beta_t \eta_{tc})} \right] \tag{4.5}$$

The quantities $n_{mt}^{\neg i}$ and $n_{mtc}^{\neg i}$ again correspond to n_{mt} and n_{mtc} but excluding the instance i. The complexity of updating the $\boldsymbol{z}$ variables is $O(NMTC)$.

Even thought the algorithm is completely unsupervised, i.e. the value of the ground truth $\boldsymbol{Z}$ is unobserved, if this information is available for a subset of the instances, we can easily incorporate it to the model. We just have to fix the ground truth of those instances to the observed value and sample z_i for the remaining ones using Equation 4.5.

Finally, we sample the concentration parameter α following the procedure proposed by Escobar (1994) (see also Appendix C).

Once the Gibbs sampler converges, we obtain samples from the posterior distribution of the unobserved variables given the annotation, i.e. $p(\boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{\alpha}|\boldsymbol{Y})$. Then, if we want to get an estimation of the confusion matrix of each cluster, we can sample the variables $\boldsymbol{\Psi}^m$ from the following set of conditional distributions that we obtain by uncollapsing Equation 4.4:

$$\boldsymbol{\Psi}_t^m|\text{rest} \sim \text{Dir}(\beta_t \boldsymbol{\eta_t} + \boldsymbol{n_{mt}})$$

where $\boldsymbol{n_{mt}} \in \mathbb{R}^C$ is a vector whose c^{th} element is n_{mtc}. In this way, the samples of the parameters $\boldsymbol{\Psi}^m$ have much lower variance than if they were drawn from the joint state space. A similar approach can be follow to estimate τ.

4.2.3.2 hcBCC

As in the cBCC we start by integrating out the $\boldsymbol{\Psi}^\ell$ and $\boldsymbol{\tau}$ variables:

$$p(\boldsymbol{Y}|\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\beta}) = \prod_m \prod_{\ell:q_\ell=m} \prod_t \left[\frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t \eta_{tc}^m)} \right],$$
$$p(\boldsymbol{z}|\epsilon,\boldsymbol{\mu}) = \frac{\Gamma(\epsilon)}{\Gamma(N+\epsilon)} \prod_t \frac{\Gamma(n_m+\epsilon\mu_t)}{\Gamma(\epsilon\mu_t)}$$

The variables we need to sample from are $\boldsymbol{\pi}$ and the ground truth estimate $\boldsymbol{z}$. Note however that we cannot marginalize out the cluster parameters $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$, as the Dirichlet prior and the Gamma prior are not conjugate to the likelihoods given above, so that these variables will have to be sampled as well.

The conditional distribution of $p(q_\ell = m|rest)$ when $m \in \boldsymbol{\pi}^{\neg\ell}$ can be computed like in the cBCC model. However, to compute $p(q_\ell = m|rest)$ when $m = \emptyset$ we need to integrate the parameters of the new clusters, i .e . $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$. In this case, due to the non-conjungancy we cannot solve this integral analytically. Instead, we use the recently proposed *Reuse algorithm* (Favaro and Teh, 2012). This algorithm is similar to the well-known Algorithm 8 (Neal, 2000), where the idea is to use a set of h auxiliary empty clusters H_{empty} to approximate the integral. However, the *reuse algorithm* is more efficient as it requires less simulations from the prior over the cluster parameters. For each cluster $m \in \boldsymbol{\pi} \cup H_{empty}$ we keep track of the parameters $\boldsymbol{\beta}^m$ and $\boldsymbol{\eta}^m$. The conditional distribution of q_ℓ is then:

$$p(q_\ell = m|\text{rest}) \propto \begin{cases} n_m^{\neg\ell} \times \prod_t \frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t^m \eta_{tc}^m)}, & m \in \boldsymbol{\pi}^{\neg\ell} \\ \frac{\alpha}{h} \times \prod_t \frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t^m \eta_{tc}^m)}, & m \in H_{empty} \end{cases}$$

If an auxiliary empty cluster is chosen, it is moved into the partition $\boldsymbol{\pi}$, and a new empty cluster is created in its place by sampling from the prior over cluster parameters. If a cluster in $\boldsymbol{\pi}$ is emptied as a result of sampling q_ℓ, it is moved into H, displacing one of the empty clusters (picked uniformly at random). In addition, at regular intervals the parameters of the empty clusters are refreshed by simulating them from their priors, while those in $\boldsymbol{\pi}$ are updated. The complexity of updating the $\boldsymbol{q}$ variables is $O(LMTC)$.

Again, due to the non-conjugacy of the Dirichlet and Gamma priors, the conditional distributions of the parameters $\boldsymbol{\eta}^m$ and $\boldsymbol{\beta}^m$ for $m \in \boldsymbol{\pi}$ cannot be computed analytically. To solve this, we use an auxiliary variable method similar to the one proposed by Escobar (1994) and Teh et al. (2003). Specifically, we introduce two auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ (see Appendix B), and apply the following Gibbs updates that leave invariant the posterior distribution:

$$\nu_{\ell t} \sim \text{Beta}(\beta_t^{q_\ell}), \qquad s_{\ell tc} \sim \text{Antoniak}(n_{\ell tc}, \beta_t^{q_\ell}\eta_{tc}^{q_\ell})$$

$$\eta_{t:}^m \sim \text{Dir}\left(\sum_{\{\ell:q_\ell=m\}} s_{\ell t:} + \phi_t \gamma_{t:}\right),$$

$$\beta_t^m \sim \text{Gamma}\left(\sum_{\{\ell:q_\ell=m\}}\sum_c s_{\ell tc} + a_t, b_t - \sum_{\{\ell:q_\ell=m\}} \log(\nu_{\ell t})\right)$$

Here the Antoniak distribution (Antoniak, 1974) is simply the distribution of the number of clusters in a partition of $n_{\ell tc}$ items under a CRP with concentration parameter $\beta_t^{q_\ell}\eta_{tc}^{q_\ell}$.

To update z_i, we compute its conditional distribution given the rest of the variables:

$$p(z_i = t|\text{rest}) \propto \left(n_t^{\neg i} + \epsilon\mu_t\right) \prod_m \prod_{\{\ell:q_\ell=m\}} \frac{\prod_c (n_{\ell tc}^{\neg i} + \beta_t \eta_{tc})^{\mathbb{I}(y_{i\ell}=c)}}{(n_{\ell t}^{\neg i} + \beta_t)^{\mathbb{I}(y_{i\ell}\neq 0)}} \tag{4.6}$$

If z_i is observed to a subset of the instances, we fix the ground truth of those instances to the observed value and sample z_i for the remaining ones using Equation 4.6.

The complexity of updating the $\boldsymbol{z}$ variables is $O(NLTC)$. Finally, we use the same scheme as the one applied in Section 4.2.3.1 to update α.

Like in 4.2.3.1, once our Gibbs sampler converges we get samples from the posterior distribution of the unobserved variables, i.e. $p(\boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{Y})$, just by discarding the values of the auxiliary parameters, which is equivalent to integrate them out. Finally, we can infer the individual confusion matrix corresponding to each annotator just sampling from the following set of conditional distributions:

$$\boldsymbol{\Psi}_t^\ell|\text{rest} \sim \text{Dir}(\beta_t^m \boldsymbol{\eta}_{\boldsymbol{t}}^m + \boldsymbol{n}_{\boldsymbol{\ell t}})$$

where $\boldsymbol{n}_{\boldsymbol{\ell t}} \in \mathbb{R}^C$ is a vector whose c^{th} element is $n_{\ell tc}$.

4.2.4 Related Work

Ghahramani and Kim (2003) and Kim and Ghahramani (2012) proposed a method called Independent Bayesian Combination of Classifiers (iBCC), whose graphical model is shown in Figure 4.1. In this model, each annotator is characterized by a unique confusion matrix Ψ_ℓ and they are independent given the ground truth and the properties of the annotators. The estimation of each confusion matrix relies on the information we have about the corresponding annotator. If the annotator has provided too few annotations so far, we have a poor estimation that can harm the performance of the algorithm.

Ghahramani and Kim (2003) and Kim and Ghahramani (2012) also presented two extensions. The first one uses a latent variable that categorizes the instances in two classes: easy and difficult to classify. The assumption is that the annotators have the same behavior regarding the easy instances, while they are different for the difficult ones. In the second one, they propose a more flexible correlation model based on a factor graph. However, these models do not identify groupings of annotators.

In our cBCC model, if α tends to infinity, every annotator is allocated to a singleton cluster, and it becomes equivalent to the iBCC model. We see that in this case, the correlation a priori among two annotators (Equation 4.3) tends to zero. By reducing the value of α we increase the value of the correlation between the different annotators, i.e. the annotators start to be grouped in clusters sharing the parameters that govern their behavior. For $\alpha = 0$ all annotators are allocated to the same cluster an the model assumes every annotator has the same behavior, being equivalent to majority voting.

Notice that the vector $\boldsymbol{\beta}^m$ governs the variability among the annotators that belong to the same cluster m. The bigger are these values, the lower is the intra-cluster variability, i.e. the probability mass of the corresponding Dirichlet distribution gets more concentrated around its mass. If we make each of the components of $\boldsymbol{\beta^m}$ tend to infinity, then the variability among the annotators tend to 0 and the model becomes equivalent to the cBCC model forcing all the annotators of the

cluster to share the exact same confusion matrix. In this way, this model can be seen as a generalization of some state-of-the-art methods (see Section 4.2.4).

Regarding the hcBCC, we saw in Section 4.2 that if we make each of the components of $\boldsymbol{\beta^m}$ tend to infinity, then the variability among the annotators tend to 0 and the model becomes equivalent to the cBCC. Also, in the hcBCC model, if each component of $\boldsymbol{\phi}$ tends to infinity, and we also make the quantities a_t and b_t tend to infinity with a fixed $\frac{a_t}{b_t}$ ratio for all t, then we recover the iBCC model with $\boldsymbol{\eta}_t = \boldsymbol{\gamma}_t$ and $\beta_t = \frac{a_t}{b_t}$. If α tends to ∞, then the model is equivalent to the iBCC model, but with additional priors on $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$. To sum up, we can see each the cBCC and the iBCC as particularizations of the hcBCC model, which capture more complex relationships among the annotators.

Simpson et al. (2011) extend the proposal of Ghahramani and Kim (2003) and Kim and Ghahramani (2012) in two directions. First, they derived a variational inference algorithm for the iBCC, which is more efficient for large datasets. Second, they apply community detection algorithms to the estimated confusion matrices to detect clusters of annotators with the same behavior. Recently, they have extended the model to the case in which the properties of the annotators can vary in time (Simpson et al., 2013). In both cases, the detection of groups of annotators is made in a post-hoc manner and therefore, this information is not used to improve the estimation of the confusion matrices of the annotators or the ground truth estimate. To the extent of our knowledge, only Kajino et al. (2013) perform the inference of the groups of annotators and the ground truth at the same time, using convex optimization. However, the performance depends on a constant that controls the strength of the clustering and for tuning this constant, the authors rely on a labeled validation set. Our algorithm, on the other hand, is fully unsupervised and therefore can be apply to the standard problem presented by Ghahramani and Kim (2003) and Kim and Ghahramani (2012). Recently, Venanzi et al. (2014) published a similar model. It also takes as reference the iBCC, but they use a a different observation model based on a softmax and they select the number of communities using the marginal likelihood and performing

line-search on a prefixed range of values.

Recently, a paper on the inconsistency of the DP Mixture Model to estimate the true number of components was published (Miller and Harrison, 2013). However, we are not interested in estimating the "true" number of annotators' clusters, specially since this is not a well defined measure in a real crowdsourcing application. Instead, we look for identifying a clustering of the annotators that improves the performance and helps us to better understand the different types of annotators that are present in the crowdsourcing application.

4.3 Experiments

In this section, we firstly use synthetic datasets based on different assumptions to validate our models. In the second part we use publicly available real datasets to compare our models with state-of-the-art algorithms highlighting their advantages.

4.3.1 Synthetic datasets

We generate three different datasets following respectively the assumptions of the hcBCC, cBCC and iBCC models. In order to analyze the properties of the algorithms, we apply each of them to each of the generated datasets.

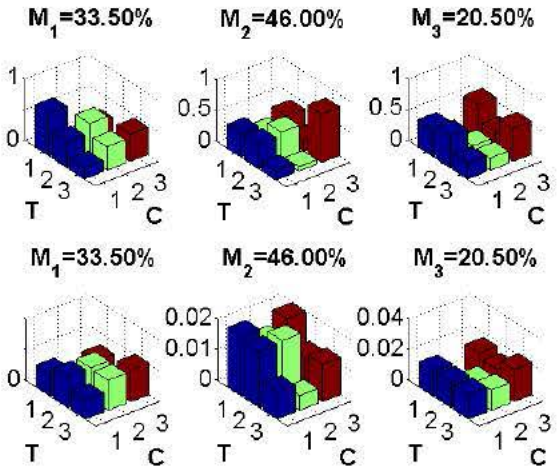
Firstly, we generate a synthetic database called dataset1 following the generative model for the hcBCC. This dataset has 500 labeled instances provided by 200 annotators. The number of categories is $C = 3$. These annotators belong to 3 clusters with properties shown in Figure 4.4a, where we can see the mean of each cluster, their variances and the percentage of annotators allocated to each of them.

We analyze the behavior of the different algorithms with respect to the sparsity of the input matrix $\boldsymbol{Y}$. In particular, we randomly erase a percentage of the entries from 82.5% missing entries to 97.5% in steps of 2.5%. This high sparsity levels are typical in crowdsourcing applications, where the idea is to distribute the load of labeling a dataset among many annotators, and therefore each of them only labels a small subset of the dataset.

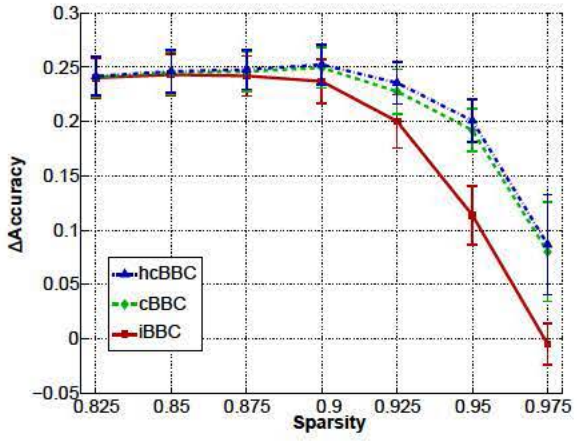
In the iBCC, the diagonal elements of $\boldsymbol{\eta}$ are set to 0.7 while the off diagonal are 0.3, which reflect our prior belief that annotators perform better than random. All the elements of $\boldsymbol{\beta}$ are 3. In the cBCC model, the hyper parameters of α are $a_\alpha = 1$ and $b_\alpha = 10$. This values agree with our prior belief that if the annotations are very scarce, simpler algorithms like majority voting are more suitable and therefore, we should favor partitions with a small number of clusters. For these parameters, in the limiting case when the sparsity of $\boldsymbol{Y}$ tends to 100%, the average number of clusters tends to 1. Finally, in the hcBCC model, we set $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$ to the values of $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ in the cBCC model respectively. All the components of a_t are set to 30 while all the components of b_t are set to 2. This reflects our prior belief that the variability among the annotators inside the clusters should be less than the variability across clusters.

We run the MCMC for 10,000 iterations. After 3,000 we collect 7,000 samples to compute $\boldsymbol{z}$ and $\boldsymbol{\pi}$. In the cBCC and hcBCC, we set to five the number of iterations to sample α following the algorithm proposed by Escobar (1994). In the hcBCC we fix the number of auxiliary clusters of the *Reuse Algorithm* to $h = 10$.

The increment in accuracy of ours proposals and the iBBC algorithm with respect to majority voting is shown in Figure 4.4b. The two proposed models outperform iBCC as expected. This improvement of both methods cBCC and hcBCC is particularly significant when the level of sparsity is high, which is a situation that we face in the early stages of a crowdsourcing project. In this case there is not enough information to accurately estimate the confusion matrix of every annotator independently. We can see that the performance of iBBC drops below the performance of the majority voting algorithm, which assumes all annotators are similar. Therefore, identifying a clustering structure that allows to share some parameters among the annotators helps to increase the accuracy of the estimates. Notice that the performance obtained by Simpson et al. (2011) would be equal to the performance of the iBBC model given that it identifies the annotators' clusters after the ground truth has been estimated, so it does not affect the performance of the algorithm.

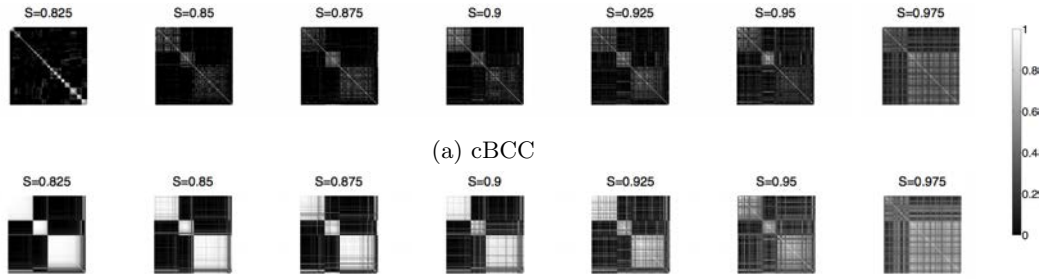


(a) annotators' properties for dataset1. (Upper row) Mean of the clusters. (Lower row) Variance of the clusters.



(b) Performance for dataset1

Figure 4.4: Results for dataset1. a) Characteristics of the annotators' clusters present in dataset1. M_i denotes the percentage of annotators allocated to cluster i, T is the ground truth label, and C is the annotator label. b) Results for dataset 1. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels.



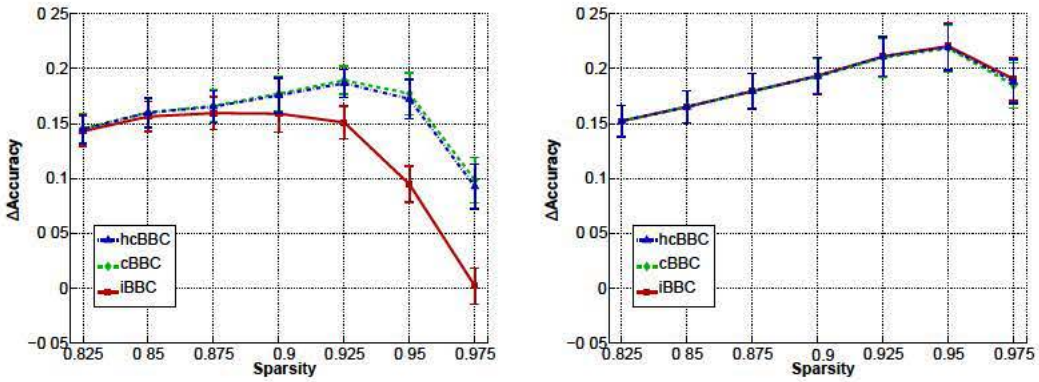
(a) cBCC

(b) hcBCC

Figure 4.5: Co-occurency matrices of the annotators

To further analyze the cluster structure identified by the algorithms, in Figure 4.5 we represent the co-occurrence matrix of the annotators. The position (ℓ, ℓ') is the probability of ℓ and ℓ' belonging to the same cluster. We can see that the clusters identified by the hcBCC are more useful than the ones extracted by the cBCC, because in a practical situation we are not normally interested in finding annotators with exactly the same behavior, but annotators with similar characteristics. For example, we can see that when 82.5% of the annotations are missing, the hcBCC algorithm identifies the 3 main groups of annotators while the cBCC algorithm identifies instead a much larger number of groups because of the constraint that all annotators of a cluster must have the same properties. So, although both algorithms' performance is similar, the clustering provided by the hcBCC is easier to interpret and gives a simpler explanation of the data.

Finally, we test with datasets that are generated following the iBCC and cBCC models. First, we create a new dataset (dataset2) in which the mean confusion matrix of each cluster is the same as in dataset1 which is shown in Figure 4.4a. However, in this case the variability of the confusion matrices inside each cluster is zero. Therefore, this new dataset follows the assumptions made by the cBCC model. Again, the performance of the cBCC and the hcBCC models outperforms iBCC as expected (see Figure 4.6a). However, even though data is generated from the cBCC which is a simpler model than the hcBCC, hcBCC is able to discover the underlying structure of the annotators and gets a performance which is on par



(a) Performance for dataset2

(b) Performance for dataset3

Figure 4.6: Results for dataset2 and dataset3. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels

with the cBCC. The hcBCC does not degrade the solution although it is more flexible.

In the last database called dataset3, we generate all the instances from the same clustering (M_2 in Figure 4.4a). In this case there is no different clusters of annotators and each of them has its own confusion matrix. Therefore, this dataset fulfill the assumptions of the iBCC. In Figure 4.6b we see that the performance of the two proposed models is identical to iBCC. To sum up, we see that the performances of cBCC and hcBCC dominate iBCC under all conditions tested.

4.3.2 Real datasets

In this section, we use 4 publicly available crowdsourced datasets with $C = 2$ whose principal characteristics are described in Table 4.1 (Raykar and Yu, 2012).

To choose the hyper-parameters we follow the reasoning of Section 4.3.1. Specifically, in the iBCC the diagonal elements of $\boldsymbol{\eta}$ are 0.7 while the off diagonal are 0.3, and all the elements of $\boldsymbol{\beta}$ are set to 3. In this way, we incorporate our prior belief that annotators are imperfect but perform better than chance. In the cBCC model we use the same value for $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ so that the comparison is fair. Finally for the hcBCC model, γ is set to the same value used for $\boldsymbol{\eta}$ in the previous models.

Dataset	N	L	μ_n	μ_l	Sparsity (%)	Brief Description
bluebird	108	39	108	39	0	Identify wether there is a Indigo Bunting or Blue Grosbeak in the image
rte	800	164	49	10	93.90	Identify wether the second sentence can be inferred from the first
valence	100	38	26	10	73.68	Identify the overall positive or negative valence of the emotional content of a headline
temp	462	76	61	10	86.84	Annotators observe a dialogue and two verbs from the dialogue and have to identify whether the first verb event occurs before or after the second

Table 4.1: Description of the real datasets. N and L denotes the number of instances and annotators respectively. μ_n stand for the mean number of instances labeled by a annotator and μ_l designate the mean number of annotators that label an instance.

All the components of a_t are set to 20 while all the components of b_t are set to 2, reflecting our belief that the variability inside clusters should be lower than the variability across clusters. We fix $a_\alpha = 1, b_\alpha = 10$ in both, cBCC and hcBCC. We run the MCMC for 10,000 iterations and we discard the first 3,000 to compute the posterior distribution of $\boldsymbol{z}$ and $\boldsymbol{\pi}$.

In Table 4.2, we see the performance of the different algorithms in terms of accuracy predicting the ground truth. In particular, we see that the performances of the cBCC and the hcBCC are better than that of the iBCC in the last three datasets, i.e. rte, temp and valence. On the other hand, in the bluebird datasets the iBBC performs better. Notice again that the performance of the algorithm described by Simpson et al. (2011) would be exactly equal to the one of the iBCC, given that the communities of annotators are inferred after the ground truth is inferred and therefore, it does not affect the accuracy in any way.

The performance difference between the cBCC and the hcBCC is only significant in the valence dataset. However, the main advantage of the hcBCC model over the cBCC is clear when we represent the average number of clusters (see

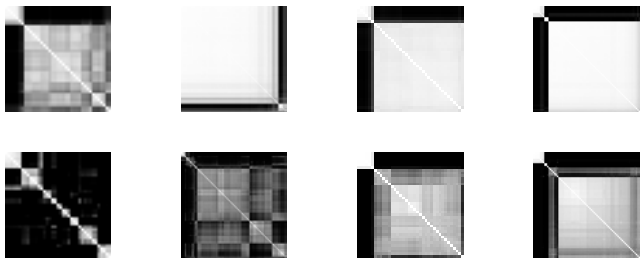
Dataset	Accuracy(%)				Average number of clusters	
	Majority	iBCC	cBCC	hcBCC	cBCC	hcBCC
bluebird	75.93	**89.81**	88.89	88.89	11.32 ± 0.04	3.31 ± 0.09
rte	91.88	92.88	**93.12**	**93.12**	7.70 ± 0.07	2.30 ± 0.06
valence	80.00	85.00	88.00	**89.00**	3.5 ± 0.04	2.25 ± 0.02
temp	93.94	94.35	**94.37**	**94.37**	6.20 ± 0.03	3.2 ± 0.02

Table 4.2: Results for the real data. Mean accuracy of the different algorithms [1]. Average number of clusters (mean ± one standard deviation).

Figure 4.7 and Table 4.2). Even though the cBBC model correctly captures the clustering structure of the annotators, forcing all annotators of a cluster to share the same confusion matrix translates into a large number of clusters, some of them with very similar properties.

The hcBCC identifies a smaller number of clusters that are much more interpretable, in the sense that it perfectly identifies each kind of clusters thanks to its additional flexibility. We are not interested in identifying clusters of annotators with the exact same behavior, but what we really want is to find clusters of annotators that behave in a similar way, so we can establish strategies to boost the overall performance of the crowdsourcing system, i.e. by rewarding the most efficient labelers, avoiding spammers or by better defining the description of the task based on the biases identified in the clusters of annotators.

In Figure 4.8 we show as an example the mean confusion matrix of the hcBCC clusters in the datasets. It shows very interpretable clusters that are useful for the modeler. In the bluebird dataset we can clearly identify a small subset of experts ($M_4 = 15.38\%$) who shows a high performance labeling the bird images. In addition, we find that the biggest cluster ($M_2 = 35.90\%$) corresponds to annotators whose accuracy is high when the real class is $z = 1$ (images of Blue Grosbeak) but performs poorly when the class is $z = 2$ (images of Indigo Bunting). Finally, we have two clusters of spammers. In the first cluster ($M_1 = 15.38\%$) annotators tend to label all images as belonging to class $z = 2$ and in the second ($M_3 = 33.33\%$)



(a) bluebird (b) rte (c) valence (d) temp

Figure 4.7: Co-ocurrency matrix of the annotators. (Upper row) hcBCC. (Lower row) cBCC.

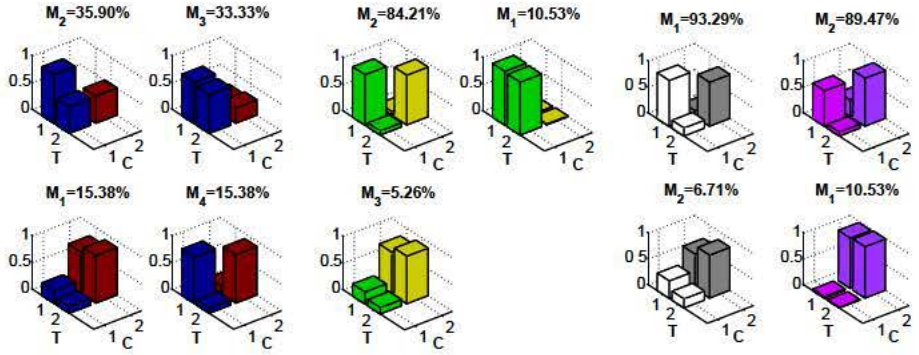
annotators tend to label all images as $z = 1$. In the temp dataset, we can observe that the majority of the annotators ($M_2 = 84.21\%$) are experts, but there are again two small clusters of spammers.

As for the rte dataset, most of the annotators have a good performance ($M_1 = 93.29\%$). The remaining annotators are bias toward labeling instances as belonging to class $z = 2$. Finally, in the valence dataset we can see that the majority of the annotators ($M_2 = 89.47\%$) are very accurate identifying instances belonging to class $z = 2$ and have a medium performance when $z = 1$. In addition we find a small cluster of annotators that have labeled almost every instance as $z = 2$. All this information about the underlying clustering structure of the annotators in the datasets can be used in a real crowdsourcing application to develop efficient strategies to minimize the cost of a crowdsourcing project maximizing the performance.

To conclude this section, we evaluate the performance of the algorithms for different levels of sparsity. Following the procedure in Section 4.3.1, we create 50 random databases for each level of sparsity. We do that in such a way that every instance has at least one label and every annotator provides at least one label. The results are shown in Figure 4.9.

In the dataset bluebird and temp, we observe that finding clusters of annotators

[1]The standard deviations are less than 10^{-4} and are not shown

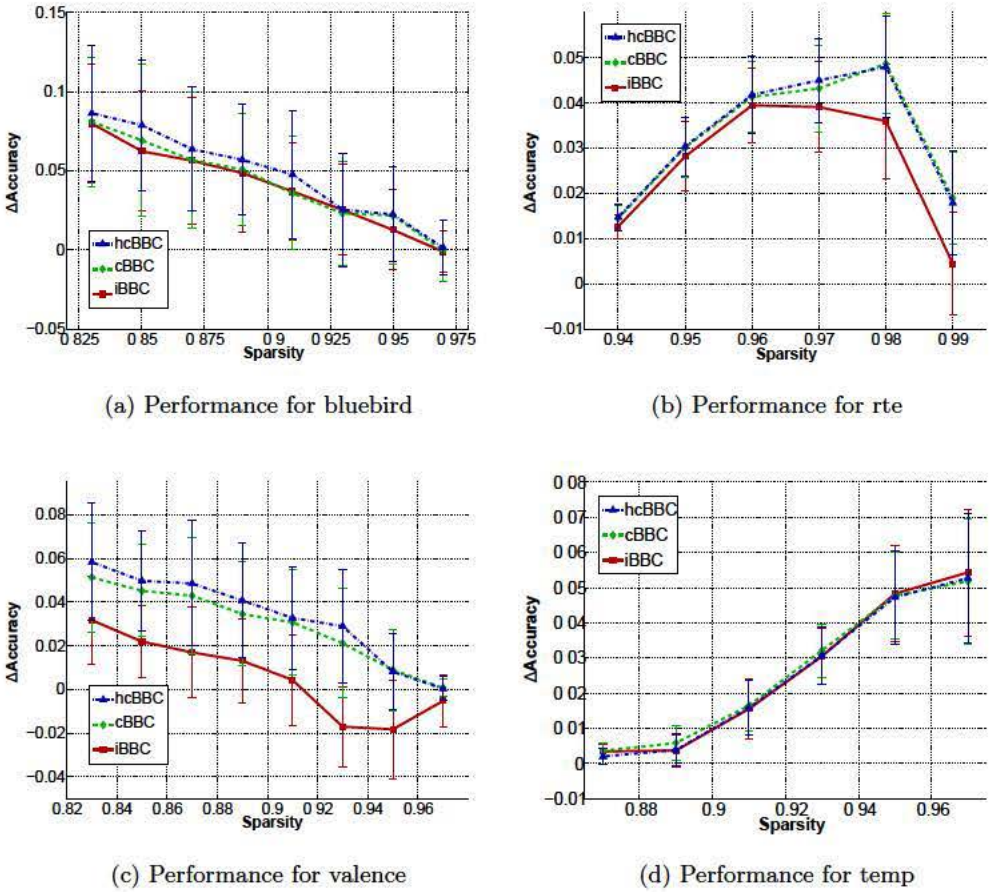


(a) bluebird (b) temp (c) rte (d) valence

Figure 4.8: Mean confusion matrices of the annotator's clusters identified by hcBCC. M_i denotes the percentage of annotators allocated to cluster i, T is the ground truth label, and C is the annotator label.

does not have a significant effect in terms of accuracy. However, the cBCC and the hcBCC models do not degrade the performance and give us some insight about the annotators in the crowdsourcing application (see Figure 4.8). In the rte and valence datasets, the inference of the clustering structure of the annotators also translates into an improvement in terms of accuracy. In the rte dataset, this improvement is not significant for the original sparsity level, but it becomes more significant when the sparsity is increased. What happens is that when the sparsity is very high, there are very few annotations provided by each annotator, and the iBBC algorithm fails to infer the properties of each annotator separately.

In the valence dataset, we can even see that the performance of the iBBC model drops below the performance of a simple majority voting algorithm when the sparsity is increased. However, the cBCC and hcBCC outperform the majority voting algorithm for every sparsity level. Again the iBBC model does not have enough information to infer the properties of each annotator and a simpler model like majority voting, which assume that all annotators have the same level of expertise, performs better. Actually, what is happening is that the the CRP prior used in the cBCC and the hcBCC models favors partitions with a small number of



(a) Performance for bluebird

(b) Performance for rte

(c) Performance for valence

(d) Performance for temp

Figure 4.9: Results of the iBCC, cBCC and hcBCC for real datasets. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels

clusters. When the input matrix $\boldsymbol{Y}$ is very sparse, the prior term dominates over the likelihood and all annotators tend to be grouped in the same cluster.

4.4 Conclusions

In this chapter, we have proposed two new BNP models to merge the information provided by the annotators in a crowdsourcing system. The main idea is to capture the underlying clustering structure in the annotators space, helping to improve the ground truth estimate, specially when the matrix of annotators is highly sparse.

In addition, the identified clustering structure is a valuable information by itself that can be use to devise crowdsourcing strategies that minimize the overall cost.

In the cBCC model, we have used a CRP to infer the partitioning of the annotators such that annotators in the same cluster are constrained to have the same properties. In the hcBCC model, we have used a hierarchical structure to increase the flexibility. In particular, each annotator has its own properties, but annotators assigned to the same cluster have similar properties. In this way, it finds smaller number of clusters that are easy to interpret.

We have shown how these new models relate to the iBCC model and analyzed the correlation structure among the annotators as a consequence of the clustering. We have proposed MCMC methods to infer the parameters of both models and performed several experiments with synthetic and real databases, which have shown that the algorithms outperform the current state-of-the-art. These results can be also found in G. Moreno et al. (2014).

5

Modeling Inconsistent Annotators

5.1 Introduction

In Chapter 4, we proposed algorithms to combine the annotations $y_{i\ell}$ in an optimal way to recover z_i. In particular, we focused our efforts on scenarios in which the annotation matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times L}$ was highly sparse, i.e. each annotator only labels a small subset of the instances. In this chapter, we focus on a rather different scenario. In particular, we consider the situation in which we have a small number of annotators that have labeled nearly all instances in the dataset, and we want to combine these annotations to get a better estimation of the ground truth than the one provided individually by each of them. In these scenarios we have enough information to model the variability in the behavior of the annotators when they label instances belonging to different areas of the covariate space.

The need for varying sensitivity and specificity can be understood, as annota-

tors might be expert annotators in some cases and novice in others. For example, the success of an annotator solving an image labeling task may depends on several factors like his past experience annotating similar images, the intrinsic difficulty of the image, the attributes of the images that he pays attention to and other personal biases that he may have acquired. Due to these factors, his performance might be in-homogeneous across the instance space. Furthermore, this information can be used to extract the areas in which each annotator does best, as well as the most informative attributes for the images in each of the identified areas to improve the accuracy and minimize the cost.

In this chapter, we propose a generative method using a Dirichlet Process (DP) prior (Ferguson, 1973) to cluster those areas across which the annotators are consistent, using a similar approach to that used in mixture of experts (Yuksel et al., 2012). Unlike discriminative methods like Raykar et al. (2010), generative learning allows explaining to annotators how the final decision is taken and helping them to improve their decision process. Moreover, it allows generating synthetic examples to validate the achieved model (Ulusoy and Bishop, 2005). Based on this DP prior, we build a Bayesian Nonparametric (BNP) model that automatically adjusts the number of clusters depending on the structure of the data. Moreover, our model is inductive , allowing us to directly infer a classifier for future unseen instances (see chapter 2).

In Della Penna and Reid (2012) the authors have proved that, without any assumption about the behavior of the annotators, it is impossible to learn . We focus on the applications where the annotators are experts in the task and we can safely assume that their sensitivity and specificity are above chance. In this situation, it is possible to learn without knowing the ground truth. Moreover, if the ground truth is known for some instances, our method can accommodate that information. In addition, our model provides an intuitive solution, showing how good is each annotator in all areas of the instances space.

The chapter is organized as follows. In Section 5.2 we present the model and its variational inference to find the posterior distribution of its parameters, as well

as the predictive distribution. In Section 5.2.4 we review the previous and related work in the field of multiple annotator learning. In Section 5.3, experimental results are shown to illustrate its validity and evaluate its performance compared to other state-of-the-art algorithms. Finally, Section 5.4 concludes the chapter.

5.2 Bayesian Combination of Non-Homogeneous Annotators

We advocate for a generative modeling of the annotators, because we are not only interested in achieving the state-of-the-art performance but also in understanding the underlying mechanism by which each annotator makes its prediction. So the results in the model can be used as a computer-aided diagnosis, as well as to advise about which expert should be consulted in each case and help those with poorer results to understand their limitation and bias to improve their future performances. From this point of view we have designed a BNP model fixing a DP prior over the areas of the instance space under which the annotators are consistent. This amounts to assuming an initial infinite number of areas and finding a suitable number of them depending on the complexity of the data.

The Bayesian Combination of Non-Homogeneous Annotators (BCNHA) model takes into account all the information given by the different annotators in order to jointly infer the number of components, the parameters of the annotators in each component, the ground truth for the training set an a classifier for future unlabeled instances. Modeling all this quantities together can lead to an improvement of the overall performance or the system as we can use the estimation of the regions where the annotators are consistent to improve the estimation of the ground truth and vice versa.

5.2.1 Model

We consider the case of binary classification for simplicity, i.e. binary ground truth $z \in \{0, 1\}$ and binary annotations $\boldsymbol{y} \in \{0, 1\}^L$, although it can be readily extended

to a multi-class classification scenario. The joint distribution of the covariates $\boldsymbol{x} \in \mathbb{R}^D$ and the response variables $\boldsymbol{y}$ is decomposed as $p(\boldsymbol{x}, \boldsymbol{y}) = \sum_z p(\boldsymbol{y}|z)p(z|\boldsymbol{x})p(\boldsymbol{x})$ explicitly modeling the conditional distribution of the actual label given the instance, i.e. $p(z|\boldsymbol{x})$.

We rely on a Mixture Model (MM) to represent the joint distribution of the instances $p(\boldsymbol{x}, \boldsymbol{y})$, i.e. the distribution over the joint space of the covariates and the annotation. Regarding the marginal distribution $p(\boldsymbol{x})$ we represent it by a Gaussian Mixture Model (GMM) as they are general enough to fit any input distribution, as well as easy to adjust and it uses local information to cluster nearby inputs. We assume that for each cluster the annotators are consistent. This assumption is based on believing that each annotator performs similarly to similar inputs.

The graphical model is shown in Figure 5.1. Given a set of examples labeled by a set of L annotators $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$ where $y_{i\ell}$ represent the label given by the ℓ^{th} annotator to the instance i, we assume the following observation model:

$$\boldsymbol{x}_i|q_i, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K} \sim \mathrm{N}(\boldsymbol{\mu}_{q_i}, \boldsymbol{\Sigma}_{q_i}), \tag{5.1}$$

$$z_i|\boldsymbol{x}_i, \boldsymbol{w}_{1:K}, b_{1:K}, q_i \sim \mathrm{Bern}(\sigma(\boldsymbol{w}_{q_i}^T x_i + b_{q_i})), \tag{5.2}$$

$$y_{i\ell}|q_i, z_i, \boldsymbol{\alpha}_{1:K}, \boldsymbol{\beta}_{1:K} \sim \begin{cases} \mathrm{Bern}(\alpha_{q_i\ell}), & \text{if } z_i = 1 \\ \mathrm{Bern}(1 - \beta_{q_i\ell}), & \text{if } z_i = 0 \end{cases} \tag{5.3}$$

where $\sigma(s) = 1/(1+\exp(-s))$ is the sigmoid function. The variable $q_i \in \{1 \dots K\}$ identifies the component from which the sample $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ comes from. In particular, the conditional distribution of $\boldsymbol{x}_i$ given $q_i = k$ follows a Gaussian distribution with mean $\boldsymbol{\mu}_k \in \mathbb{R}^D$ and covariance $\boldsymbol{\Sigma}_k \in \mathcal{Z}^{D\times D}$, where is the space of positive semi-definite and symmetric $D \times D$ matrices.

Given the component $q_i = k$ the classifier that relates the instances $\boldsymbol{x}_i$ and the ground truth z_i is given by a logistic regression with parameters $\boldsymbol{w}_k \in \mathbb{R}^D$ and $b_k \in \mathbb{R}$. Although the classification boundary is locally linear given the cluster, the global classification boundary is nonlinear, while preserving the interpretability of the solution.

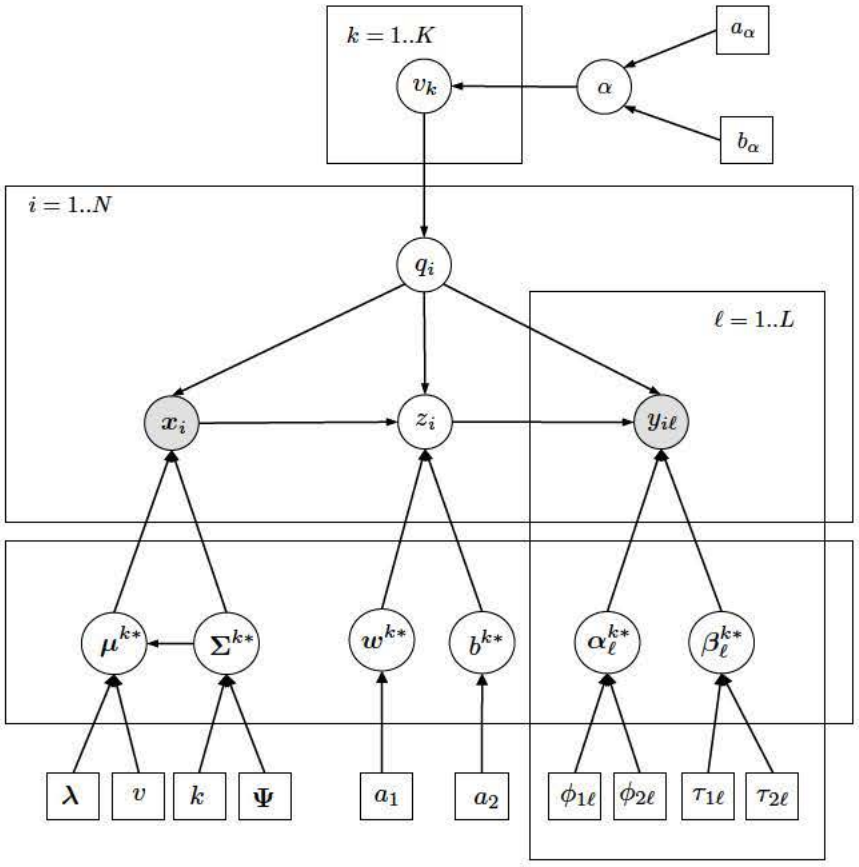


Figure 5.1: Graphical model for BCNHA model, where the top box represents the prior on the number of clusters, the middle box represents the observations and the ground truth as a latent variable and the bottom box represents the parameters of the clusters.

For the instances that belongs to the same component $q_i = k$, we assume that the behavior of the annotators is uniform and therefore, each of them is characterized by a sensitivity $\alpha_{k\ell} \in [0, 1]$ and a specificity $\beta_{k\ell} \in [0, 1]$. In this way, each annotator is no longer characterized by a unique sensitivity and specificity, but by a set of them that model de differences in its behavior for different areas of the instances space, i.e. we flexibilize the model to accommodate locally homageneous annotators.

In addition, we assume that the annotators are independent given q_i and z_i. Given these assumptions, the distribution of $\boldsymbol{y}_i$ given the component q_i and the

ground truth z_i follows a multiplicative binomial distribution :

$$\boldsymbol{y_i} \sim a^{z_i} b^{1-z_i}, \tag{5.4}$$

$$a_i = \prod_{\ell=1}^{L} (\alpha_{q_i \ell})^{y_{i\ell}} (1 - \alpha_{q_i \ell})^{1-y_{i\ell}}, \quad b_i = \prod_{\ell=1}^{L} (\beta_{q_i \ell})^{1-y_{i\ell}} (1 - \beta_{q_i \ell})^{y_{i\ell}}. \tag{5.5}$$

The problem is that the inferences are sensitive to the number of components K chosen. In the limiting case $K = 1$ the model assumes that the annotators are homogeneous across the whole instance space. If K is too large the model is to complex and we are not able to infer accurately the properties of the annotators. In the limiting case $K = N$, each user has a different behavior for each instance and therefore, no learning is possible.

To solve the issue, we resort to BNP (see chapter 3). The BCNHA starts from a Dirichlet Process Mixture Model (DPMM) Antoniak (1974). Specifically, we rely on a Stick Breaking (SB) construction Sethuraman (1994) because it is a suitable representation to apply variational inference:

$$v_k|\alpha \sim \text{Beta}(1, \alpha), \quad \pi_k = v_k \prod_{j=1}^{k-1} (1 - v_j), \tag{5.6}$$

$$q_i|\boldsymbol{\pi} \sim \text{Discrete}(\boldsymbol{\pi}). \tag{5.7}$$

Here, $\alpha \in \mathbb{R}_+$, $v_k \in [0, 1]$ and $\pi \in \mathcal{S}^\infty$, where $\mathcal{S}^\infty$ is the infinite dimensional probability simplex (See chapter 3). In this way, the BCNHA has a priori infinitely many components, from which only a finite subset of them have a non zero weight a posteriori.

The SB prior tends to favor MM with a small number of component. This is sensible in our case, in which it makes sense to assume a priori that the annotators are homogeneous across the instance space. However, once we see the data, the model automatically infer the necessary number of components depending on the complexity of the dataset.

We define the following prior distributions for the hyper-parameters,

$$\boldsymbol{\mu}_k, \boldsymbol{\Sigma_k}|\boldsymbol{\lambda}, v, p, \boldsymbol{\Psi} \sim \mathrm{NW}^{-1}(\boldsymbol{\lambda}, v, p, \boldsymbol{\Psi}), \tag{5.8}$$

$$\alpha_{k\ell}|\phi_{1\ell}, \phi_{2\ell} \sim \mathrm{Beta}(\phi_{1\ell}, \phi_{2\ell}), \tag{5.9}$$

$$\beta_{k\ell}|\tau_{1\ell}, \tau_{2\ell} \sim \mathrm{Beta}(\tau_{1\ell}, \tau_{2\ell}), \tag{5.10}$$

$$\boldsymbol{w}_k \sim \mathrm{N}(0, a_1^2 \boldsymbol{I}_{D\times D}), \quad b_k \sim \mathrm{N}(0, a_2), \tag{5.11}$$

where NW^{-1} represent a Normal Inverse Wishart distribution. Except for the parameters of the final local classifiers $\boldsymbol{w}_k, b_k$, we have chosen conjugate priors which are desirable for simplifying the inference of the parameters of the model. We also define a gamma distribution over the concentration parameter of the DP i.e. $\alpha|\chi_1, \chi_2 \sim \mathrm{Gamma}(\chi_1, \chi_2)$, as was suggested in West et al. (1994) to make the model more robust with respect to the concentration parameter.

5.2.2 Inference

We are interested in the posterior distribution of the latent or hidden variables

$$\Theta = \{q_{1:N}, z_{1:N}, \boldsymbol{v}_{1:\infty}, \boldsymbol{\mu}_{1:\infty}, \boldsymbol{\Sigma}_{1:\infty}, \boldsymbol{\alpha}_{1:\infty}, \boldsymbol{\beta}_{1:\infty}, \bar{\boldsymbol{w}}_{1:\infty}\},$$

given the data

$$\mathcal{D} = \{x_{1:N}, y_{1:N}\}$$

and the hyperameters

$$\mathcal{H} = \{\boldsymbol{\lambda}, v, p, \boldsymbol{\Psi}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2, a_1, a_2, \chi_1, \chi_2\}.$$

Here we denote by $\bar{\boldsymbol{w}}_k$ the concatenation of the weight vector $\boldsymbol{w}_k$ and the bias b_k. However, getting the exact posterior distribution $p(\Theta|\mathcal{D}; \mathcal{H})$ is computationally intractable as we need to compute its normalization constant which implies to integrate over all possible values of the latent variables, i.e. $p(\mathcal{D}; \mathcal{H}) = \int p(\mathcal{D}|\Theta; \mathcal{H}) p(\Theta; \mathcal{H}) d(\Theta)$. Therefore, we need to resort to approximate inference methods.

Since the proposal of the DPMM, approximate inference schemes based on Markov Chain Monte Carlo (MCMC) methods have played a crucial role. Instead,

we use variational inference to speed up the convergence when we work with large datasets (See chapter 3).

In our case, we assume that the variational distribution that we use to approximate the posterior distribution factorizes in the following way:

$$g_\nu(\boldsymbol{\theta}) = g_{\nu_1}(q_{1:N}) g_{\nu_2}(z_{1:N}) g_{\nu_3}(\boldsymbol{v}_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K}, \boldsymbol{\alpha}_{1:K}, \boldsymbol{\beta}_{1:K}, \bar{\boldsymbol{w}}_{1:K}), \tag{5.12}$$

where we have truncated it making $v_K = 1$ as was proposed by Blei and Jordan (2006). This truncated process closely approximate a true DP when when the truncation level K is chosen large relative to the number of data points.

We need to choose a distribution for each of these factors, as well as the values of their parameters ν to maximize the following lower bound of the evidence of the model:

$$\ln(\mathcal{D}; \mathcal{H}) \geq E_{g_{(\boldsymbol{\nu})}}\{\ln(p(\Theta, \mathcal{D}; \mathcal{H}))\} - E_{g_{(\boldsymbol{\nu})}}\{\ln(g_{\boldsymbol{\nu}}(\Theta))\}. \tag{5.13}$$

For a particular subset of latent variables $\boldsymbol{\theta}_n$, the factor $g(\boldsymbol{\theta}_n)$ that maximizes the bound in Equation 5.13, is given by the following expression, where $\boldsymbol{\theta}_{-n}$ represents the remaining latent variables:

$$g(\theta_n) \propto \exp(E_{g(\theta_{-n})}\{\ln(p(\Theta, \mathcal{D}; \mathcal{H}))\}). \tag{5.14}$$

The complete log-likelihood $log(p(\Theta, \mathcal{D}; \mathcal{H}))$ has the following form in our case:

$$\begin{aligned}
\ln(p(\Theta, \mathcal{D}; \mathcal{H})) = &\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{I}(q_i = k)\Big(z_i\Big\{\ln(\sigma(\bar{w}_k^\top \bar{\boldsymbol{x}}_i) + \ln(a_i)\Big\} \\
&+ (1 - z_i)\Big\{\ln(\sigma(-\bar{w}_k^\top \bar{\boldsymbol{x}}_i) + \ln(b_i)\Big\} + \ln(\mathrm{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) \\
&+ \ln(v_k) + \sum_{j=1}^{k-1}\ln(1 - v_j)\Big) + \sum_{k=1}^{K}\Big\{\ln(\mathrm{NW}^{-1}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma_k}|\boldsymbol{\lambda}, v, p, \boldsymbol{\Psi})) \\
&+ \ln(\mathrm{N}(\bar{\boldsymbol{w}}_k|0, \boldsymbol{C})) + \sum_{\ell}[\ln(\mathrm{Beta}(\alpha_{k\ell}|\phi_{1\ell}, \phi_{2\ell})) \\
&+ \ln(\mathrm{Beta}(\beta_{k\ell}|\tau_{1\ell}, \tau_{2\ell}))]\Big\} + \sum_{k=1}^{K-1}\ln(\mathrm{Beta}(v_k|1, \alpha)) + \ln(\mathrm{Gamma}(\alpha|\chi_1, \chi_2)),
\end{aligned} \tag{5.15}$$

where $\bar{\boldsymbol{x}}_i$ are the feature vectors with an intercept, i.e. $\bar{\boldsymbol{x}}_i = [\boldsymbol{x}_i^\top 1]^\top$, $\mathbb{I}(.)$ represents the indicator function and $\boldsymbol{C} = \text{diag}[a_1^2, a_1^2, \ldots, a_1^2, a_2^2]$ is a matrix of dimension $(D+1) \times (D+1)$. Given the factorization in Equation 5.12 we can update the parameters in an iterative way giving rise to a coordinate ascent algorithm.

1. Obtain initial guesses for $g_{\nu_1}(q_{1:N})$, $g_{\nu_2}(z_{1:N})$ and $g_{\nu_3}(\boldsymbol{v}_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K}, \boldsymbol{\alpha}_{1:K}, \boldsymbol{\beta}_{1:K}, \bar{\boldsymbol{w}}_{1:K})$

2. Update $g_{\nu_1}(q_{1:N})$ according to:

$$g_{\nu_1}(q_{1:N}) \propto \exp\left\{\mathbb{E}_{g_{\nu_2} g_{\nu_3}}(\ln(p(\Theta, \mathcal{D}; \mathcal{H})))\right\}. \tag{5.16}$$

3. Update $g_{\nu_2}(z_{1:N})$ according to:

$$g_{\nu_2}(z_{1:N}) \propto \exp\left\{\mathbb{E}_{g_{\nu_1} g_{\nu_3}}(\ln(p(\Theta, \mathcal{D}; \mathcal{H})))\right\}. \tag{5.17}$$

4. Update $g_{\nu_3}(\boldsymbol{v}_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K}, \boldsymbol{\alpha}_{1:K}, \boldsymbol{\beta}_{1:K}, \bar{\boldsymbol{w}}_{1:K})$ according to:

$$\begin{aligned} &g_{\nu_3}(\boldsymbol{v}_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K}, \boldsymbol{\alpha}_{1:K}, \boldsymbol{\beta}_{1:K}, \bar{\boldsymbol{w}}_{1:K}) \propto \\ &\exp\left\{\mathbb{E}_{g_{\nu_1} g_{\nu_2}}(\ln(p(\Theta, \mathcal{D}; \mathcal{H})))\right\}. \end{aligned} \tag{5.18}$$

5. Compute the value of the lower bound in Equation 5.13 and check it to evaluate if the algorithm has converged. If that is not the case go back to the second step.

Observing Equations 5.16, 5.17 and 5.18 we can see that a fully factorized variational distribution arises:

$$\begin{aligned} q(\boldsymbol{\theta}) = &\prod_{i=1}^{N} g(q_i) \prod_{i=1}^{N} g(z_i) \prod_{k=1}^{K} g(\boldsymbol{v}_k) \prod_{k=1}^{K} g(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &\prod_{k=1}^{K} \prod_{\ell=1}^{L} g(\alpha_{kl}) \prod_{k=1}^{K} \prod_{\ell=1}^{L} g(\beta_{kl}) g(\bar{\boldsymbol{w}}_k) g(\alpha). \end{aligned} \tag{5.19}$$

Notice that the set of additional independencies that appear in Equation 5.19 when we compare it with Equation 5.12, are not the consequence of making additional assumptions. Instead, these independencies follows directly from the algorithm.

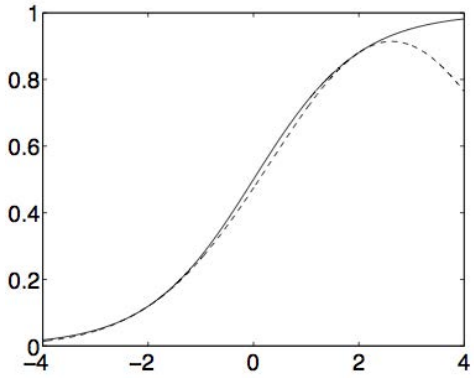


Figure 5.2: The logistic function $\sigma(\bar{\boldsymbol{w}}_k^\top \bar{\boldsymbol{x}}_i)$ (solid line) and its variational lower bound (dashed line) for $\xi = 2$. Borrowed from Jordan and Jaakkola (1996).

The sigmoid function that appear in Equation 5.15 does not allow to directly compute the updating equations, as we need to compute the expectation of a sigmoid function with respect to a gaussian density function. Instead, we use the following variational lower bound proposed by Jordan and Jaakkola (1996),

$$\sigma(\bar{\boldsymbol{w}}_k^\top \bar{\boldsymbol{x}}_i) \geq \sigma(\xi_{ik}) \exp\Big\{ \frac{\bar{\boldsymbol{w}}_k^\top \bar{\boldsymbol{x}}_i - \xi_{ik}^2}{2} + \eta(\xi_{ik}) \Big[((\bar{\boldsymbol{w}}_k^\top \bar{\boldsymbol{x}}_i)^2 - \xi_{ik}^2 \Big] \Big\}, \tag{5.20}$$

where

$$\eta(\xi_{ik}) = \frac{1}{2\xi_{ik}} \Big\{ \sigma(\xi_{ik}) - \frac{1}{2} \Big\}. \tag{5.21}$$

This lower bound has the nice property that it is at most quadratic in the quantity $\bar{\boldsymbol{w}}_k^\top \bar{\boldsymbol{x}}_i$, allowing the analytic computation of the corresponding expectations. Notice that this lower bound depends on a set of parameters $\{\xi_{ik}\}_{\{i=1:N,k=1:K\}}$, so we have to maximize the lower bound with respect to the value of these auxiliary parameters in order to make it as tight as possible. In Figure 5.2, we can visualize how this lower bound approximates $\sigma(\bar{\boldsymbol{w}}_k^\top \bar{\boldsymbol{x}}_i)$ when $\xi = 2$.

Applying this lower bound, we get the following form for the factors involved

in Equation 5.19:

$$g(q_i) = \text{Discrete}(\boldsymbol{\pi}_{1i}, \boldsymbol{\pi}_{2i}, \ldots, \boldsymbol{\pi}_{Ki}), \tag{5.22}$$

$$g(z_i) = \text{Bernoulli}(\boldsymbol{\eta}_i), \tag{5.23}$$

$$g(v_k) = \begin{cases} \text{Beta}(\kappa_{1k}, \kappa_{2k}) & k = 1 : K-1 \\ \delta(v_k - 1) & k = K \end{cases} \tag{5.24}$$

$$g(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \text{NW}^{-1}(\boldsymbol{\lambda}_k, v_k, p_k, \boldsymbol{\Psi}_k), \tag{5.25}$$

$$g(\alpha_{kl}) = \text{Beta}(\sigma_{1kl}, \sigma_{2kl}), \tag{5.26}$$

$$g(\beta_{kl}) = \text{Beta}(\epsilon_{1kl}, \epsilon_{2kl}), \tag{5.27}$$

$$g(\bar{\boldsymbol{w}}_k) = \text{N}(\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k), \tag{5.28}$$

$$g(\alpha) = \text{Beta}(s_1, s_2). \tag{5.29}$$

The updating equations for the parameters and the analytical form of the lower bound are given in Appendix D. The coordinate ascend algorithm is guaranteed to converge to a local maximum of the likelihood function. This lower bound allows assessing when the algorithm has converged as well as compare the result of the algorithm with different initializations in order to choose the one that is closer to the global maximum of the the marginal likelihood.

5.2.3 Predictive distribution

Apart from the inference of the parameters of the model, we are also interested in predicting the ground truth of new instances coming from the same distribution. We compute the following predictive distribution

$$p(z^*|\boldsymbol{x}^*, \mathcal{D}; \mathcal{H}) = \int p(z^*|\boldsymbol{x}^*, \Theta, \mathcal{D}; \mathcal{H}) dP(\Theta|\mathcal{D}; \mathcal{H}). \tag{5.30}$$

Given the factorization of the posterior, we can approximate it by the following expression:

$$\begin{aligned} p(z^*|\boldsymbol{x}^*, \mathcal{D}; \mathcal{H}) &= \frac{1}{\sum_{k=1}^{K} E_{g_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}} \{p(\boldsymbol{x}^*|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\} E_{g_{v_{1:K}}} \{\boldsymbol{\pi}(v_{1:K})\}} \\ &\times \sum_{k=1}^{K} \Big(E_{g_{\bar{\boldsymbol{w}}_k}} \{p(z^*|\boldsymbol{x}^*, \bar{\boldsymbol{w}}_k)\} E_{g_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}} \{p(\boldsymbol{x}^*|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\} E_{g_{v_{1:K}}} \{\boldsymbol{\pi}(v_{1:K})\} \Big), \end{aligned} \tag{5.31}$$

where:

$$E_{g_{v_{1:K}}}\{\boldsymbol{\pi}(v_{1:K})\} = \frac{\kappa_{1k}}{\kappa_{1k}+\kappa_{2k}}\prod_{j=1}^{k-1}\frac{\kappa_{2j}}{\kappa_{1j}+\kappa_{2j}}, \tag{5.32}$$

$$E_{g_{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k}}\{p(\boldsymbol{x}^*|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\} = \frac{1}{\pi^{D/2}}\frac{\Gamma_D\left(p_k+1/2\right)}{\Gamma_D\left(p_k/2\right)}\frac{|\boldsymbol{\Psi}_k|^{p_k/2}}{|\boldsymbol{\Psi}'|^{p_k+1/2}}\Big(\frac{v_k}{v_k+1}\Big)^{D/2}, \tag{5.33}$$

$$\Psi'_k = \Psi_k + \frac{v_k}{v_k+1}(\boldsymbol{x}^*-\boldsymbol{\lambda}_k)(\boldsymbol{x}^*-\boldsymbol{\lambda}_k)^\top. \tag{5.34}$$

Here we have used $\Gamma_D(.)$ to denote the multivariate gamma function. To compute the expectation with respect to $g_{\bar{\boldsymbol{w}}_k}$ in Equation 5.31 we need to resort again to the approximation in Equation 5.20. Applying this approximation and developing the expression we get the following:

$$E_{g_{\bar{\boldsymbol{w}}_k}}\{p(z^*=1|\boldsymbol{x}^*,\bar{\boldsymbol{w}}_k)\} \geq max_{\xi_k^*}\frac{\sigma(\xi_k^*)|\widehat{\boldsymbol{B}}_k|}{|\widehat{\boldsymbol{\Sigma}}_k|}\times\exp\Big(-\frac{1}{2}\Big\{\widehat{\boldsymbol{\mu}}_k^\top\widehat{\boldsymbol{\Sigma}}_k\widehat{\boldsymbol{\mu}}_k - \widehat{\boldsymbol{e}}_k^\top\widehat{\boldsymbol{B}}_k\widehat{\boldsymbol{e}}_k\Big)\Big\} \tag{5.35}$$

$$\widehat{\boldsymbol{B}}_k = \Big(\widehat{\boldsymbol{\Sigma}}_k + 2\eta(\xi_k^*)\bar{\boldsymbol{x}}^*\bar{\boldsymbol{x}}^{*\top}\Big)^{-1}, \tag{5.36}$$

$$\widehat{\boldsymbol{e}}^\top = \Big(\widehat{\boldsymbol{\mu}}_k^\top\widehat{\boldsymbol{\Sigma}}_k^{-1} + \frac{\bar{\boldsymbol{x}}^*}{2}\Big)\widehat{\boldsymbol{B}}_k \tag{5.37}$$

Once we have approximated the predictive distribution we just need to apply a threshold to estimate the label of the test example.

5.2.4 Related work

To the extent of our knowledge, only two inductive methods have considered a varying sensitivity and specificity across the instance space for the annotators. In Yan et al. (2010) the authors propose a discriminative model which assumes that the annotators are independent and that the label of each of them is corrupted by Gaussian additive noise. The model considers in-homogeneous annotators by making the noise variance to depend on the particular instance following a sigmoid function. This parametrization is somewhat rigid and it cannot be easily interpreted to, for example, help physicians improve their diagnosis. In Zhang and Obradovic (2011), the authors assume that the data comes from a gaussian mixture model (GMM) and that the behavior of the annotators is uniform in each

component. To fit the GMM they discard the labeling, which contains relevant information to determine the areas across which the annotators are consistent. They use the Bayes Information Criterion (BIC) approximation Fraly and Raftery (1998) to fix the number of components. After this first step, they apply the algorithm in Raykar et al. (2010) to infer the sensitivity and specificity of the annotators in each of the identified areas.

The final classifier of the BCNHA is similar to other BNP supervised algorithms like Shahbaba and Neal (2009); Hannah et al. (2010); Wang et al. (2010). However these methods have been designed to solve a traditional supervised task and therefore, they can not deal with multiple annotated training sets. In addition, there are also differences in the method used to infer the parameters. In particular Shahbaba and Neal (2009); Hannah et al. (2010) propose a method based on MCMC. In Wang et al. (2010) a variational method is used, but they used a different classification function for the local classifiers (a cumulative Gaussian distribution).

Another research line consists of analyzing whether it is possible to learn only from the labels provided by the annotators, without knowing the ground truth. It has been proved that without any assumptions about the annotators, it is not possible to learn the ground truth as there may be other equally likely solutions that makes the problem ambiguous Della Penna and Reid (2012). In Wauthier and Jordan (2011), the authors defend that observing the ground truth for some instances, allows to avoid this ambiguity. They also propose a model in which each annotator is modeled by a linear classifier. To contemplate the correlation among the annotators these linear classifiers are a linear combination of an infinite set of latent variables with an Indian buffet process as a prior.

5.3 Experimental results

5.3.1 Synthetic dataset

A synthetic database has been built to show that the BCNHA model works well under the model's assumptions. To that end, we generate a set of two dimensional instances coming from a finite mixture distribution with five components. Then, we assign three of them to the positive class, i.e. $z = 1$, and the remaining two to the negative class, i.e. $z = 0$.

To simulate the labels given by each of the annotators, we are going to assume that each of them makes its decision based on a different view. A view is a subset of the total number of attributes D that defines the instances $\boldsymbol{x}$. We represent the indexes of the attributes contained in the view used by the annotator ℓ by $\mathcal{I}_\ell$ and the view of the annotator ℓ of a particular instance $\boldsymbol{x}$ by $\boldsymbol{x}^{d:d\in\mathcal{I}_\ell}$. In this way, each annotator bases his decision on partial information.

This example mimics the way a physician would evaluate a patient, from a large set of tests each physician takes only a subset that might give him sufficient information to evaluate a patient, at an affordable price. Depending on how many tests they ask for, and what particular tests they ask for, their sensibilities and specificities vary.

Three annotators are simulated in this way. In particular, the first annotator uses both attributes ($\mathcal{I}_1 = \{1, 2\}$) while the second annotator only the first attribute ($\mathcal{I}_2 = \{1\}$) and the third annotator only the second attribute ($\mathcal{I}_3 = \{2\}$).

To compute their labels we use a simple maximum likelihood estimator using the particular view selected for each annotator. For each annotator $\ell \in \{1, \ldots, L\}$, we compute the sample mean $\boldsymbol{\mu}_{\ell k} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i^{d:d\in\mathcal{I}_\ell}$ and the sample covariance matrix $\boldsymbol{\Sigma}_{\ell k} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i^{d:d\in\mathcal{I}_\ell} - \boldsymbol{\mu}_{\ell k})^\top(\boldsymbol{x}_i^{d:d\in\mathcal{I}_\ell} - \boldsymbol{\mu}_{\ell k})$ of each cluster. Then, we generate the labels of the annotator ℓ by allocating each instance to the component that maximizes the likelihood, i.e. choosing the k that maximizes $N(\boldsymbol{x}_i^{d:d\in\mathcal{I}_\ell}|\boldsymbol{\mu}_{\ell k}, \boldsymbol{\Sigma}_{\ell k})$, and labeling the instance with the label associated with the chosen component.

Following this scheme, we generate $N = 1000$ samples and we randomly select

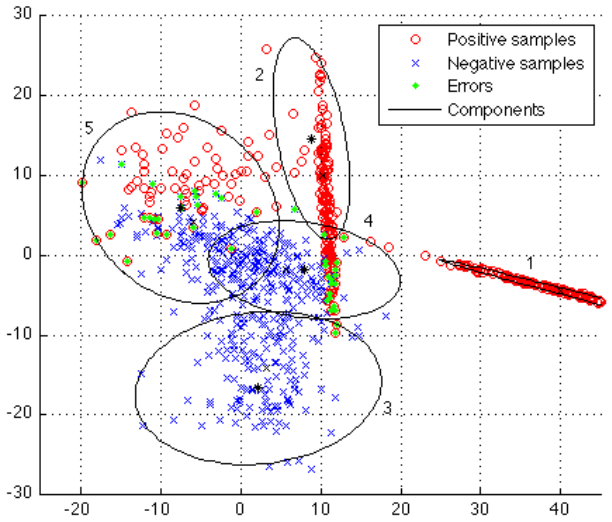


Figure 5.3: Output of the BCNHA for the simulated database

80% of the samples as our training set and test the performance of the algorithm with the remaining 20%. We use the training set to infer the parameters of the model, i.e. mean, variance, ground truth and characteristics of each annotator, for each component using the inference algorithm described in Section 5.2.2, where we use a truncated the variational approximation with 30 components, i.e. $K = 30$. Once we have obtained the parameters of the model, we apply the final classifier to the test set to validate the model. In Figure 5.3, we can see the sub-groups or clusters that are identified by the BCNHA model, as well as the ground truth inferred for the samples that belongs to each of these clusters.

The true and estimated sensitivity and specificity of each of the clusters identified by the BCNHA model are shown in Tables 5.1 and 5.2. Notice that, whenever there are enough samples the estimation is quite close to the real value. For example in component two, there are enough positive samples to estimate the sensitivity. However, if there are few samples, the estimation is skewed toward the prior. For example, in component three there are only three positive samples, and therefore the real value of the sensitivity is not significant. This is sensible given that in clusters with a small number of samples of one class, the sample estimate of the parameters of the annotators is not a good estimator. In this cases, maximum

Table 5.1: Real Sensitivity/Specificity

Cluster ID	1st	2nd	3th
1	100/−	100/−	72.02/−
2	98.23/100	86.73/100	99.11/0
3	0/100	0/90.34	0/100
4	71.43/98.17	83.67/92.68	26.53/60.37
5	79.36/90.43	57.14/19.15	93.65/48.94

Table 5.2: Estimated Sensitivity/Specificity

Cluster ID	1st	2nd	3th
1	98.02/88.23	97.9/88.24	74.66/88.27
2	96.35/89.49	87.68/88.27	96.53/87.73
3	88.36/97.73	88.39/89.4	88.17/97.43
4	91.88/97.89	91.35/88.18	53.32/64.02
5	93.56/96.96	71.05/47.73	95.75/63.1

likelihood methods tend to overfit to the data.

Averaging the results over 100 repetitions, in which we randomly select the training and test sets, we get the results shown in Table 5.4. We compare the performance of the BCNHA with two state-of-art algorithms Yan et al. (2010); Zhang and Obradovic (2011). To make a fair comparison with Yan et al. (2010) and Zhang and Obradovic (2011), we impose the following constraint $\{\boldsymbol{w}_k\}_{k=1}^{K} = \boldsymbol{w}$ forcing the final classifier to be linear. In addition, we show the performance of the model without such constraint, i.e. allowing a non-linear classifier. Finally, we show the performance of a logistic regression using a majority voting strategy $\widehat{z_i} = \sum_{\ell=1}^{L} y_{i\ell}$.

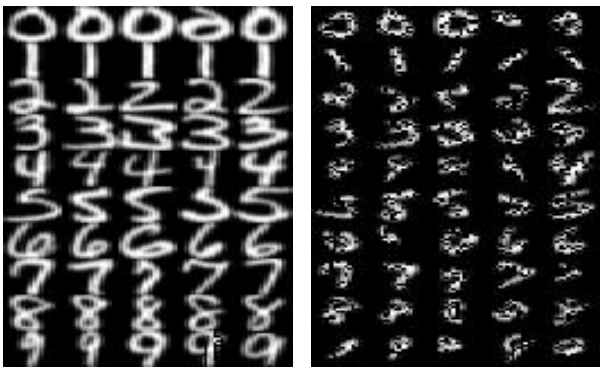
5.3.2 Real Datasets

In this section we have tested the BCNHA model using two real datasets. The first one is the well-known USPS handwritten 16x16 digits dataset. This dataset

consists of 11000 8-bit grayscale images. There are 1100 examples per class, being the classes the digits from 0 to 9. For our purpose we randomly select 1000 examples and we tackle the binary task of identifying whether a digit is even or odd. The idea is to create a dataset in which the performance of the annotators depends on other latent factors apart from the ground truth. The obvious example that comes to mind is the number itself, i.e. an annotator may have a different performance classifying a image of a 2 than classifying an image of a 4 as even numbers. However, there may be other not so obvious latent factors that determine the performance of the annotator, e.g. an annotator may perform better in an area of the instance space which contains thin strokes images or numbers with significant smaller sizes. It is important to note that the BCNHA model identifies the different areas of the instance space without being explicitly told about which factors characterize these different areas. It just performs a clustering in the product space of the covariates and the annotations.

This subset is evaluated by 5 different annotators with different characteristics (age, gender, visual handicaps...). All the annotators evaluate all the cases. In addition, as the accuracy of humans in handwritten digits is 98.26% Dong (2001), the samples presented to the annotators are not the actual digits, but a distorted version of them. This distortion is random and independently applied to each of the samples. Moreover, they are only able to see each of the digits for one second, which contributes to make the task more challenging. A small sample of both, the original dataset, as well as the distorted one are shown in Figures 5.4a and 5.4b.

Once the annotators have provided the labels, we apply PCA as a standard preprocessing step to reduce the dimensionality of the instances. We keep the first 20 principal components, retaining 90% of the total variance of the signal, and we randomly select 80% of the samples for training leaving the remaining 20% for testing. We approximate the posterior with a truncated variational distribution with 30 components, i.e. $K = 30$, inferring the parameters from the labels provided by the annotators (without using the ground truth). We train the BCNHA model using the training samples with the labels provided by the annotators (without



(a) (b)

Figure 5.4: (a) Original samples from the USPS database. (b) Distorted samples from the USPS database. A random distortion that includes a rotation, a partial occlusion and a polynomial deformation is applied to the samples before being presented to the annotators.

using the ground truth). The experiment is repeated 100 times and the results are shown in Table 5.4. We observe that the BCNHA shows the best performance. We see that the variant with the nonlinear classifier have a better performance due to the nonlinearity of the task. However, the linear variant also outperforms the other algorithms. The algorithm in Yan et al. (2010) also shows a good performance, but the algorithm in Zhang and Obradovic (2011) fails to capture the complexity of the database due to the use of a less robust model selection strategy only based on the covariates. In the BCNHA model, we infer a posterior distribution over the space of all possible partitions of the instances, based not only on the covariates but on the annotations. In addition, to estimate the ground truth of the training set, the BCNHA averages over the posterior distribution of all possible partitions, which is more robust to over-fitting than selecting a fix number of components by solving a maximization problem.

In Figure 5.5, we can see the 9 largest clusters found by the BCNHA model. Each column represents a different cluster. They are arranged from left to right according to their sizes, being the largest one represented by the first column on the left. We can see that some of these clusters reflect common mistakes committed by the annotators. For example, in cluster 2 we can see that annotators are prone

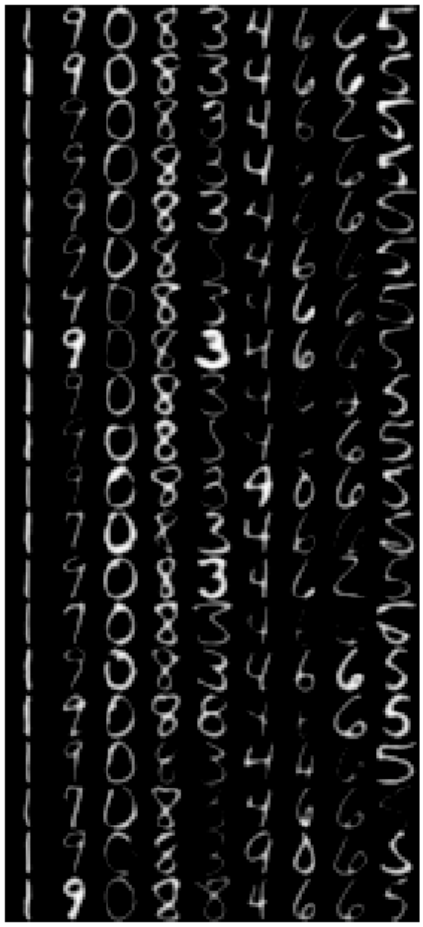


Figure 5.5: Identified clusters. Each column shows the 20 most representative instances of a cluster. The clusters are ordered according to their sizes. The first column on the left represents the largest cluster.

to commit mistakes between numbers 7 and 9, although this is not counted as an error in the current task of discriminating between even and odd numbers. In clusters 5, we can observe similarities between the numbers 3, 5 and 8. In cluster 6 annotators are going to make mistakes between 4 and 9. The BCNHA model takes advantage of this heterogeneity of the database by allowing the annotators to have different properties (sensitivity and specificity) in each cluster. Finally, in Table 5.3 we show the sensitivities and specificities of the annotators in these 9 components.

The second dataset was built based on the Electromagnetic Articulography

Table 5.3: Estimated Sensitivity/Specificity (USPS dataset)

Comp.	1st Ann.	2nd Ann.	3rd Ann.	4th Ann.	5th Ann
1	98.99/75.15	96.98/74.86	91.97/75.29	98.00/75.84	94.97/72.94
2	85.97/68.55	87.12/79.31	81.50/63.89	89.42/56.22	85.46/85.51
3	74.97/98.40	74.98/98.38	74.95/95.28	75.01/75.85	75.02/85.85
4	67.00/93.81	56.60/87.90	85.93/71.48	57.18/85.74	66.76/72.38
5	95.30/87.49	91.00/51.38	64.98/75.91	93.40/39.81	94.91/87.41
6	66.89/87.89	82.67/96.22	67.73/73.40	66.72/84.24	51.08/81.83
7	75.00/95.69	75.00/95.69	75.00/95.67	75.00/91.37	75.00/87.70
8	74.92/96.46	74.93/96.42	75.05/89.36	74.95/85.80	75.04/78.25
9	87.42/53.04	87.69/21.10	51.95/58.68	87.63/21.00	87.75/16.29

(EMA) Database Lee et al. (2005). The EMA database is a Multi-modal database used for emotion recognition. In this chapter, we use only the speech signal. Emotional speech recognition is a field whose goal is to develop algorithms that are able to identify the emotional state of a human being from its voice. Emotions recognition helps to improve the performance of human-computer interactions systems and therefore it is an important research line in the machine learning area. Some of the applications that we can find in the literature are the development of call centers software that is able to identify the frustration of the customer and therefore modify its behavior according to it, the design of systems that helps the physicians to provide better diagnosis by being more conscious of the emotional state of the patient or the design of better voice synthesizers (see Ververidis and Kotropoulos (2006)).

The EMA database consist of 534 utterances. These utterances corresponds

Table 5.4: Test error (%)

Method / Dataset	Synthetic	EMA	USPS
Majority voting	9.35 ± 1.81	$26.11 \pm (4.31)$	8.07 ± 1.85
Algorithm in Yan et al. (2010)	9.15 ± 2.42	29.96 ± 4.20	8.01 ± 1.68
Algorithm in Zhang and Obradovic (2011)	19.08 ± 9.76	25.99 ± 3.96	14.10 ± 2.68
BCNHA Linear Classif.	8.67 ± 1.66	$\mathbf{25.34 \pm 3.39}$	7.63 ± 1.75
BCNHA Non-linear Classif.	$\mathbf{6.96 \pm 2.68}$	28.19 ± 5.01	$\mathbf{5.41 \pm 1.69}$

with 14 sentences read by three speakers under four acted emotions, i.e. neutral, angry, sad and happy. There is a native-speaker man that reads the 14 sentences and two women that read 10 sentences. Each instance is recorded five times by each actor (some of the sentences had to be removed because the audio file was damaged). Each utterance is evaluated by at least three experts which classify each of them as belonging to one of the previous emotions. Among all the utterances, we select a subset of 534 instances as the most representatives of the emotions that the actors were representing.

Following the experimental setup presented in Zhang and Obradovic (2011), we tackle the binary classification task of recognizing positive emotions, i.e. happy, neutral, versus the negative ones, i.e. sad, angry. In the same way we selected 13 statics features consisting of 12 MFCCs computed from 24 filter banks and the log energy. In addition we computed 13 delta coefficients and 13 delta-delta coefficients from the speech signal over 25 ms frames with 10 ms overlap. Finally we computed the feature-wise mean over the entire utterance. The audio files were labeled by two native speakers and three non-native speakers. These are the labels that are used by the BCNHA model in the training stage. Like in the USPS database we randomly select 80% of the sampling to infer the parameters of the variational distribution with $K = 30$, leaving the remaining 20% for testing and we repeated the experiment 100 times. The results are shown in Table 5.4, where we can show that the BCNHA models with linear classifier show the best performance. The performance of Zhang and Obradovic (2011) is similar but slightly worse which

is sensible given that it is based on similar assumptions. The difference is due to the joint inference of the components, the performance of the annotators and the classifier that is done in the BCNHA model. We see that the algorithm inYan et al. (2010) is unable to capture the behavior of the annotators in this database and the performance drops below the majority voting algorithm.

The algorithm detects 6 components. The 85% of the samples fall into the two largest components of the mixture model. In particular the first component contains 64% of the utterances recorded by the actor and 33% of the ones recorded by the actresses. The second component contains 49.3% of the utterances recorded by the actresses and 23.6% of the ones recorded by the actor. This indicates, as was previously noticed in Zhang and Obradovic (2011), that the annotators have a different performance depending on the gender of the speaker.

Table 5.5: Estimated Sensitivity/Specificity

Annotator	1st Comp.	2nd Comp.
1	90.43/71.52	85.64/87.56
2	69.15/81.11	80.40/75.58
3	87.13/91.95	89.82/93.79
4	83.19/92.71	92.63/95.18
5	85.87/69.53	96.87/72.63

In Table 5.5 we can see the performance of the annotators for the 2 main identified components. This information could be useful to designate the most suitable annotator for each particular instance. For example, we can see that the annotator 2 has a significantly lower performance than the rest of the annotators, and therefore, it might be sensible to replace her in the future. In addition, the BCNHA model discovers differences in the behavior of the remaining annotators. For example, while most of the annotators show an overall better performance in the instances that are allocated to the second component, the annotator 1 performs better in the first component when the true label is 1. This suggests that we should assign to annotators 3, 4 and 5 future instances that are similar

to the ones allocated in the component 2. Of course, more advanced strategies for the assignment of future samples can be design once all this information is unraveled. For example, we could use annotator 3 to double check the instances allocated to the component 2 and labeled as 0 by the annotator 5. Likewise, we could send the instances of the component 1 labeled as 0 by the annotator 1, to the annotator 4 given the high sensitivity of the former and the high specificity of the later. Finally, knowing all this information might allow to pair the annotators so they can help each other to solve their biases in the labeling process, and therefore improving the performance of the overall system.

5.4 Conclusion

We have proposed a new inductive classification algorithm that can work under a multiple annotator scenario in which the annotators has a non uniform sensitivity and specificity across the instance space. Our algorithm is based on a generative approach since one of our main goals was to obtain an interpretable model which helps the annotators to improve their performance and identify possible biases in their decision process. We have used a DP prior to identify the areas across which the annotators are consistent, the sensitivity and specificity of each annotator in each area, the ground truth of the training examples and a classifier to predict the ground truth of future examples. We have tested the performance of the method in two real scenarios and showed that the algorithm outperforms the state-of-art algorithms. Moreover, the generative model discovers additional information about the differences in the behavior of the different annotators. The analysis of this information is useful to design strategies to increase the overall performance of the system, e.g. identifying the best annotators to label a particular instance or pairing annotators with different biases so they can learn from each other. The results of this chapter can also be found in G. Moreno et al. (2015).

6

Conclusions and Further Work

6.1 Summary

In this chapter, we summarize the contributions of this thesis, and also describe some possible lines for future research.

In this thesis we have analyzed the use of Bayesian Nonparametric (BNP) models in the context of crowdsourcing. Specifically, we have focused on the problem of combining the labels provided by a set of annotators who face a multi-class classification problem.

As a first main contribution, we have extended the transductive model for label aggregation proposed by Heung-Nam et al. (2010). By using a Chinese Restaurant Process (CRP) prior, we have achieved three main goals:

- Identify different groups of annotators with similar properties. In a crowd-

sourcing application where we have a high number of unknown annotators, this partition provides a useful and interpretable summary of our pool of annotators. This summary can be use to devise better rewarding policies for the annotators, or to increase their performances by better designed training schemes.

- Increase the robustness of the algorithm with respect to the Cold Start Problem (CSP). The model adapts its complexity, i.e. the number of parameters, depending on the available information. A priori, it favors solutions with a low number of parameters. In the limit, all the annotators are allocated to the same cluster, being equivalent to a majority voting strategy. However, if there is enough information to discard that hypothesis, it starts to consider partitions with a larger number of clusters of annotators.

- Contemplate more complex correlation's patterns among the annotators

Specifically, we have proposed two models. While the first one (Clustering based Bayesian Combination of Classifiers (cBCC)) forces all the annotators that belong to the same cluster to have exactly the same properties, the second one (Hierarchical Clustering based Bayesian Combination of Classifiers (hcBCC)) allows each annotator to have different properties from the rest, but similar to those that have been allocated to the same cluster. We have developed Markov Chain Monte Carlo (MCMC) algorithms based on Gibbs sampling and the help of auxiliary variables to cope with the non-conjugancies present in the models.

The second main contribution is a new inductive model for combining the labeling of annotators with a non-homogeneous behavior. We have modeled the inconsistencies in the behavior of the annotators by dividing the covariate space in different areas where the annotators may exhibit different performances. By using a Mixture Model (MM) and a Stick Breaking (SB), the algorithm identifies those areas, the properties of the annotators in each of the areas, the estimates of the ground truth and a classifier to classify future unlabeled instances.

The SB reflects our prior belief that the annotators tend to be consistent. It

favors partitions of the covariate space with few clusters. However, if the training set suggests otherwise, the algorithm is able to contemplate more complex models with finer partitions of the instance space, i.e. less consistent behavior of the annotators. This information can be used to better understand the weaknesses of the different annotators in the system and try to help them to correct these weaknesses improving their performances. We develop a variational inference algorithm using a mean-field approximation and an local lower bound to address the non-conjugancies in the model.

All the proposed models were tested using synthetic and real dataset in order to evaluate their performances against state-of-the-art algorithms. These results can also be found in G. Moreno et al. (2014) and G. Moreno et al. (2015).

6.2 Future Work

Our work also suggests several paths for further research, both in the algorithmic and application sides. We provide below a list with some of the potential future research lines.

Hierarchical communities of annotators. Regarding the algorithms to identify communities of annotators, one potential improvement is to consider hierarchies of annotators instead of a flat partition. Building a hierarchical clustering is a problem that has received a an increasing attention in the last years in BNP (Ghahramani et al., 2010; Paisley et al., 2012). The challenge is to keep the computational complexity of the algorithm under control.

Smooth varying inconsistency. While we have modeled the inconsistency of the annotators by dividing the instance space in a set of areas and assuming that the annotators are consistent in each of them, an alternative is to assume that the properties of the annotators vary smoothly across the instance space. This behavior can be modeled using generalized Gaussian Processes (GPs) (Shang and Chan, 2013; Chan, 2013). The challenge of this approach is the high computational complexity of the inference. Extensive empirical simulations will be needed to

compare these two hypothesis in different real scenarios.

Nested Partition Models. Another extension, is to combine the proposed methods by using BNP models that simultaneously take into account a partition of the annotators to identify clusters of users and a partition of the instances to model non-homogeneous annotators (Kemp et al., 2006; Rodriguez and Ghosh, 2009).

Normalized Complete Random Measures. The Dirichlet Process (DP) can be seen as a particular instance of a broader family of stochastic processes called Normalized Complete Random Measures. Recently, Favaro and Teh (2012) proposed a general sampling algorithm for any prior inside this family. While the DP imposes a sometimes quite restrictive shape over the prior distribution of the size of the clusters, this extension would allow to use more flexible priors inside this family.

Features Models. While so far we have only considered clustering solutions, feature solutions allow to model the possibility of an annotator simultaneous belonging to more than one community (Griffiths and Ghahramani, 2005). In the same way, we can divide the instance space in overlapping areas.

Scalable inference algorithms. The main advantage of crowdsourcing is the possibility of distributing the labeling across a big pool of users. The sizes of the datasets that are labeled in this way are increasing every year. We therefore need inference algorithms that are scalable. Stochastic Variational Inference (Hoffman et al., 2013) has become very popular recently because the implementation is reasonably easy and allows to scalate the variational inference to big datasets. Regarding MCMC, a similar approach to scalate to big datasets was proposed by Patterson and Teh (2013). In addition, developing inference algorithms that can be parallelize across multiple machines is an interesting direction, specially given the exponential proliferation of commodity clusters and the new open-source frameworks available (White, 2012; Zaharia et al., 2010).

Alternative observation models. Extending the proposed ideas to other problems, e.g. soft labels, regression, ranking, multi-label, and developing a general

and efficient software is another potential research line.

Active Learning. In active learning, the learning algorithm is able to interactively query an oracle to obtain the labels of new data points (Settles, 2010). In a multiple annotators scenario, the oracles are the annotators. Using the information about the existing groups of users as well as their performances in the different areas of the instance space, can lead to effective active learning algorithms that can reduce the cost by querying the most appropriate annotator for each instance. It is important to notice that always selecting the annotator whose performance is better for the instance to label, may lead to poor solutions if the estimate of his performance is based on a small number of samples, i.e. trade-off exploitation-exploration.

A

Induced Correlation by the cBCC model

In the Independent Bayesian Combination of Classifiers (iBCC) model, the joint probability of two users given the ground truth is:

$$p(y_{i\ell}, y_{i\ell'}|z_i = t) = \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0))} \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0))}{\Gamma(\beta_t\eta_{tc})} \times$$
$$\times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell'} \neq 0))} \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t\eta_{tc})} = p(y_{i\ell}|z_i = t) \times p(y_{i\ell'}|z_i = t) \tag{A.1}$$

Therefore:

$$\text{corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = 0 \tag{A.2}$$

In the Clustering based Bayesian Combination of Classifiers (cBCC) model,

we have the following expression for the joint distribution of $y_{i\ell}$ and $y_{i\ell'}$:

$$p(y_{i\ell}, y_{i\ell'}|z_i = t) = \left(\frac{1}{1+\alpha}\right)\left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0) + \mathbb{I}(y_{i\ell'} \neq 0))} \times \right.$$
$$\left. \times \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0) + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t\eta_{tc})}\right] + \left(\frac{\alpha}{1+\alpha}\right)\left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0))} \times \right.$$
$$\left. \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0))}{\Gamma(\beta_t\eta_{tc})} \times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell'} \neq 0))} \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t\eta_{tc})}\right] \tag{A.3}$$

We can now compute the covariance in the following way:

$$\mathrm{cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = \mathbb{E}\{\mathbb{I}(y_{i\ell} = a)\mathbb{I}(y_{i\ell'} = b)\} - \mathbb{E}\{\mathbb{I}(y_{i\ell} = a)\}\mathbb{E}\{\mathbb{I}(y_{i\ell'} = b)\} =$$
$$= \left(\frac{1}{1+\alpha}\right)\left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(a \neq 0) + \mathbb{I}(b \neq 0))} \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(a = c, a \neq 0) + \mathbb{I}(b = c, b \neq 0))}{\Gamma(\beta_t\eta_{tc})} - \right.$$
$$\left. - \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(a \neq 0))} \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(a = c, a \neq 0))}{\Gamma(\beta_t\eta_{tc})} \times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(b \neq 0))} \prod_c \frac{\Gamma(\beta_t\eta_{tc} + \mathbb{I}(b = c, b \neq 0))}{\Gamma(\beta_t\eta_{tc})}\right] \tag{A.4}$$

Assuming that $a \neq 0$ and $b \neq 0$, and considering the cases where $a = b$ and $a \neq b$ we obtain the following equation for the covariance:

$$\mathrm{Cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i'\ell'} = b)|z_i = t) = \begin{cases} -\left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\eta_{ta}\eta_{tb} & a \neq b \\ \left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\eta_{ta}(1-\eta_{ta}) & a = b \end{cases} \tag{A.5}$$

Here we have taken into account that $\Gamma(x+1) = x\Gamma(x)$. Once we get the expression of the covariance, we divide it by the square root of the variances to get the correlation:

$$\mathrm{Corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = \frac{\mathrm{Cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b))}{\sqrt{\mathrm{Var}(\mathbb{I}(y_{i\ell} = a))\mathrm{Var}(\mathbb{I}(y_{i\ell'} = b))}} \tag{A.6}$$

It is straightforward to see that $\mathrm{Var}(\mathbb{I}(y_{i\ell} = a)) = \eta_a(1-\eta_a)$, getting the expected result.

B

Inference details for the hcBCC model

Here we derive the Gibbs updates for the parameters $\boldsymbol{\beta}^m$ and $\boldsymbol{\eta}^m$ of the Hierarchical Clustering based Bayesian Combination of Classifiers (hcBCC) model. The posterior distribution of the $\boldsymbol{\beta}^m$ and $\boldsymbol{\eta}^m$ is proportional to

$$p(\boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\pi}) \propto p(\boldsymbol{Y}|\boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{\pi}) \times p(\boldsymbol{\eta}, \boldsymbol{\beta}) = \prod_{\ell} \prod_{t} \left[\frac{\Gamma(\beta_t^{q_\ell})}{\Gamma(n_{\ell t} + \beta_t^{q_\ell})} \prod_{c} \frac{\Gamma(n_{\ell tc} + \beta_t^{q_\ell} \eta_{tc}^{q_\ell})}{\Gamma(\beta_t \eta_{tc}^{q_\ell})} \right] \times$$
$$\times \prod_{m} \prod_{t} \left[\frac{\Gamma(\phi_t)}{\prod_c \Gamma(\phi_t \gamma_{tc})} \prod_{c} (\eta_{tc}^m)^{\phi_t \gamma_{tc} - 1} \right] \prod_{m} \prod_{t} \frac{b_t^{a_t}}{\Gamma(a_t)} (\beta_t^m)^{a_t - 1} \exp(-b_t \beta_t^m) \tag{B.1}$$

We cannot compute an analytic expression for $p(\boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\pi})$ because the prior on $p(\boldsymbol{\eta}, \boldsymbol{\beta})$ is no longer conjugate of the likelihood of the observations. The idea is to include two auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ such that we can compute the joint distribution $p(\boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{\nu}, \boldsymbol{s}|\boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\pi})$. To do so, we use the following relation between

the gamma function and the Stirling numbers of the first kind denoted by S:

$$\frac{\Gamma(x+n)}{\Gamma(x)} = (x)_n = \sum_{s=0}^{n} S(n,s)(x)^s$$

Here $(x)_n$ denotes the Pochhammer symbol. Taking into account also the definition of the beta distribution we reach the following expression:

$$p(\boldsymbol{\eta}, \boldsymbol{\beta} | \boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\pi}) \propto \prod_{\ell} \prod_{t} \left[\int_0^1 \nu^{\beta_t^{q_\ell} - 1} (1-\nu)^{n_{\ell t} - 1} d\nu \prod_{c} \sum_{s=0}^{n_{\ell tc}} S(n_{\ell tc}, s) (\beta_t^{q_\ell} \eta_{tc}^{q_\ell})^s \right] \times$$
$$\times \prod_{m} \prod_{t} \left[\frac{\Gamma(\phi_t)}{\prod_c \Gamma(\phi_t \gamma_{tc})} \prod_{c} (\eta_{tc}^m)^{\phi_t \gamma_{tc} - 1} \right] \prod_{m} \prod_{t} \frac{b_t^{a_t}}{\Gamma(a_t)} (\beta_t^m)^{a_t - 1} \exp(-b_t \beta_t^m) \tag{B.2}$$

And therefore we can introduce a set of auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ such that the joint distribution is given by

$$p(\boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{\nu}, \boldsymbol{s} | \boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\pi}) \propto \prod_{\ell} \prod_{t} \left[\nu_{\ell t}^{\beta_t^{q_\ell} - 1} (1-\nu_{\ell t})^{n_{\ell t} - 1} \prod_{c} S(n_{\ell tc}, s_{\ell tc}) (\beta_t^{q_\ell} \eta_{tc}^{q_\ell})^{s_{\ell tc}} \right] \times$$
$$\times \prod_{m} \prod_{t} \left[\frac{\Gamma(\phi_t)}{\prod_c \Gamma(\phi_t \gamma_{tc})} \prod_{c} (\eta_{tc}^m)^{\phi_t \gamma_{tc} - 1} \right] \prod_{m} \prod_{t} \frac{b_t^{a_t}}{\Gamma(a_t)} (\beta_t^m)^{a_t - 1} \exp(-b_t \beta_t^m) \tag{B.3}$$

and such that:

$$p(\boldsymbol{\eta}, \boldsymbol{\beta} | \boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\pi}) = \int p(\boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{\nu}, \boldsymbol{s} | \boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\pi}) d\boldsymbol{\nu} d\boldsymbol{s} \tag{B.4}$$

From Equation B.3 it is straightforward to compute the necessary conditional distributions to implement the Gibbs sampler (See Section 4.2.3.2), where the only non-standard distribution is the Antoniak distribution:

$$s_{\ell tc} \sim \text{Antoniak}(n_{\ell tc}, \beta_t^{q_\ell} \eta_{tc}^{q_\ell})$$

We can easily sample from this distribution by using a set of Bernoulli auxiliary variables in the following way:

$$s_{\ell tcj} \sim \text{Bern}\left(\frac{\alpha}{\alpha + j + -1} \right), \quad j = 1 \dots n_{\ell tc} \tag{B.5}$$

$$s_{\ell tc} = \sum_{j} s_{\ell tcj} \tag{B.6}$$

C

Sampling the concentration parameter

In this appendix we detail the algorithm for sampling the concentration parameter in the Clustering based Bayesian Combination of Classifiers (cBCC) and Hierarchical Clustering based Bayesian Combination of Classifiers (hcBCC) models. We start from the following result by Antoniak (1974):

$$P(M|\alpha, L) = P(M|\alpha = 1, L)L!\alpha^M \frac{\Gamma(\alpha)}{\Gamma(\alpha + L)} \tag{C.1}$$

where M is the number of clusters of annotators, α is the concentration parameter and L is the number of annotators. Then, we can set a prior on the concentration parameter, so we have:

$$p(\alpha|\text{rest}) \propto p(\alpha|M, L) \propto P(M|\alpha, L)p(\alpha) \tag{C.2}$$

We can rewrite Equation C.1 in the following way:

$$p(\alpha|M, L) \propto p(\alpha)\alpha^{M-1}(\alpha + L) \int_0^1 e^{\alpha}(1 - e)^{L-1} de \tag{C.3}$$

where we have used the identity:

$$\frac{\Gamma(\alpha)}{\Gamma(\alpha+L)} = \frac{(\alpha+L)\mathrm{B}(\alpha+1,\mathrm{L})}{\alpha\Gamma(L)} \tag{C.4}$$

being $\mathrm{B}(\cdot)$ the beta function, i.e. the normalization factor of a beta distribution. So $p(\alpha|M,L)$ can be seen as the marginal distribution of $p(\alpha,e|M,L)$. Using a gamma distributed prior for the concentration parameters, i.e. $\alpha \sim \mathrm{Gamma}(a_\alpha, b_\alpha)$, we have:

$$\begin{aligned} &p(\alpha,e|M,L) \propto \alpha^{a_\alpha+M-1}\exp(-\alpha(b_\alpha-\log(e)))+ \\ &L\alpha^{a_\alpha+M-2}\exp(-\alpha(b_\alpha-\log(e))) \end{aligned} \tag{C.5}$$

We can then compute the conditional distributions $p(\alpha,e|M,L)$ and $p(\alpha,e|M,L)$ in closed form:

$$\begin{aligned} &\alpha|e,M,L \sim \pi_e\mathrm{Gamma}(a_\alpha+M, b_\alpha-\log(e))+ \\ &(1-\pi_e)\mathrm{Gamma}(a_\alpha+M-1, b_\alpha-\log(e)) \end{aligned} \tag{C.6}$$

$$e|\alpha,M,L \sim \mathrm{Beta}(\alpha+1, L) \tag{C.7}$$

where:

$$\frac{\pi_1}{1-\pi_e} = \frac{a_\alpha+M-1}{L(b_\alpha-\log(e))} \tag{C.8}$$

Finally, to sample the concentration parameter we sample from these two conditional distributions (Equations C.6 and C.7) in the Gibbs algorithm, and we discard the samples of the auxiliary variable e.

D

Variational Inference Details

This appendix contain the update equations and the lower bound for the Bayesian Combination of Non-Homogeneous Annotators (BCNHA) model.

D.1 Update equations

- Parameters of $g_(\alpha_{kl})$:

$$\sigma_{1k\ell} = \sum_{k=1}^{K} \pi_{ik}\eta_i y_{i\ell} + \phi_{1\ell},$$
$$\sigma_{2k\ell} = \sum_{k=1}^{K} \pi_{ik}\eta_i (1 - y_{i\ell}) + \phi_{1\ell}$$

- Parameters of $g(\beta_{kl})$:

$$\epsilon_{1k\ell} = \sum_{k=1}^{K} \pi_{ik}(1-\eta_i)(1-y_{i\ell}) + \tau_{1\ell},$$
$$\epsilon_{2k\ell} = \sum_{k=1}^{K} \pi_{ik}(1-\eta_i)y_{i\ell} + \tau_{1\ell}$$

- Parameters of $g(q_i)$:

$$\begin{aligned}
&\boldsymbol{\pi}_{ki} = \frac{\rho_{ki}}{\sum_{k=1}^{K}\rho_{ki}} \\
&\ln(\rho_{ki}) = -\frac{D}{2}\Big[\ln(\pi) + \frac{1}{v_k}\Big] - \frac{1}{2}\ln(|\boldsymbol{\Psi}_k|) \\
&+ \frac{1}{2}\Big((\sum_{d=1}^{D}\psi(\frac{p_k+1-i}{2}) - (\boldsymbol{x}_i - \boldsymbol{\lambda}_k)\boldsymbol{\Psi}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\lambda}_k)\Big) \\
&+ (\psi(\kappa_{1k}) - \psi(\kappa_{1k}+\kappa_{2k})) + \sum_{j=1}^{k-1}\Big(\psi(\kappa_{2k}) - \psi(\kappa_{1k}+\kappa_{2k})\Big) \\
&+ \eta_i\big\{\sum_{\ell=1}^{L} y_{i\ell}(\psi(\sigma_{1k\ell}) - \psi(\sigma_{1k\ell}+\sigma_{2k\ell})) \\
&+ (1-y_{i\ell})(\psi(\sigma_{2k\ell}) - \psi(\sigma_{1k\ell}+\sigma_{2k\ell}))\big\} \\
&+ (1-\eta_i)\big\{\sum_{\ell=1}^{L}(1-y_{i\ell})(\psi(\epsilon_{1k\ell}) - \psi(\epsilon_{1k\ell}+\epsilon_{2k\ell})) \\
&+ y_{i\ell}(\psi(\epsilon_{2k\ell}) - \psi(\epsilon_{1k\ell}+\epsilon_{2k\ell}))\big\} + \ln(\sigma(\xi_{ik})) - \frac{\xi_{ik}}{2} \\
&+ (\eta_i - \frac{1}{2})\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{\top}\widehat{\boldsymbol{x}}_i - \eta(\xi_{ik})\left[\widehat{\boldsymbol{x}}_i\left(\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}} + \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{\top}\right)\widehat{\boldsymbol{x}}_i - \xi_{ik}^2\right]
\end{aligned}$$

  where $\psi(.)$ represents the digamma function.

- Parameters of $g(\widehat{\boldsymbol{w}}_k)$:

$$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}} = \left\{\boldsymbol{C}^{-1} + 2\sum_{i=1}^{N}\pi_{ik}\eta(\xi_{ik})\widehat{\boldsymbol{x}}_i\widehat{\boldsymbol{x}}_i^{\top})\right\}^{-1},$$
$$\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}} = \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}\left(\sum_{i=1}^{N}\pi_{ik}\left(\eta_i - \frac{1}{2}\right)\right)$$

- Parameters of $g(z_i)$:

$$\begin{aligned}
\boldsymbol{\eta}_i &= \frac{\eta_{1i}}{\eta_{1i}+\eta_{2i}} \\
\boldsymbol{\eta}_{1i} &= \sum_{k=1}^{K}\big[\pi_{ik}\big\{\sum_{\ell=1}^{L} y_{i\ell}(\psi(\sigma_{1k\ell})-\psi(\sigma_{1k\ell}+\sigma_{2k\ell})) \\
&+ (1-y_{i\ell})(\psi(\sigma_{2k\ell})-\psi(\sigma_{1k\ell}+\sigma_{2k\ell}))\big\} + \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{\top}\widehat{\boldsymbol{x}}_i\big] \\
\boldsymbol{\eta}_{2i} &= \sum_{k=1}^{K}\big[\pi_{ik}\big\{\sum_{\ell=1}^{L}(1-y_{i\ell})(\psi(\epsilon_{1k\ell})-\psi(\epsilon_{1k\ell}+\epsilon_{2k\ell})) \\
&+ y_{i\ell}(\psi(\epsilon_{2k\ell})-\psi(\epsilon_{1k\ell}+\epsilon_{2k\ell}))\big\}
\end{aligned}$$

- Parameters of $g(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$:

$$\begin{aligned}
N_k &= \sum_{i=1}^{N}\pi_{ik}, \boldsymbol{x}_k = \frac{1}{N_k}\sum_{i=1}^{N}\pi_{ik}\boldsymbol{x}_i \\
S_k &= \frac{1}{N_k}\sum_{i=1}^{N}\pi_{ik}(\boldsymbol{x}_i-\boldsymbol{x}_k)(\boldsymbol{x}_i-\boldsymbol{x}_k) \\
\boldsymbol{\lambda}_k &= \frac{N_k\boldsymbol{x}_k+v\boldsymbol{\lambda}}{N_k+v}, v_k = v+N_k, p_k = p+N_k \\
\boldsymbol{\Psi}_k &= \boldsymbol{\Psi}+N_kS_k+\frac{vN_k}{v+Nk}(\boldsymbol{x}_k-\boldsymbol{\lambda})(\boldsymbol{x}_k-\boldsymbol{\lambda})
\end{aligned}$$

- Parameters of $g(v_k)$:

$$\begin{aligned}
\kappa_{1k} &= \sum_{i=1}^{N}\pi_{ik}+1, \\
\kappa_{2k} &= \sum_{i=1}^{N}\sum_{j=k+1}^{K}\pi_{ij}+\frac{s_1}{s_2}
\end{aligned}$$

- Parameters of $g_\alpha(\alpha)$:

$$\begin{aligned}
s_1 &= \chi_1+K-1, \\
s_2 &= \chi_2-\sum_{k=1}^{K-1}(\phi(\kappa_{2k})-\phi(\kappa_{1k}+\kappa_{2k}))
\end{aligned}$$

- Update equation of ξ_{ik}:

$$\xi_{ik} = \widehat{\boldsymbol{x}}_i^{\top}\left(\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}+\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{\top}\right)\widehat{\boldsymbol{x}}_i \tag{D.1}$$

D.2 Lower bound

$$
\begin{aligned}
&\mathcal{LB} = E_{g_{\boldsymbol{\nu}}}\{\ln(p(\Theta, \mathcal{D}; \mathcal{H}))\} - E_{g_{\boldsymbol{\nu}}}\{\ln(g_{\boldsymbol{\nu}}(\Theta))\} \\
&= \sum_{i=1}^{N} \left\{ \ln\left(\sum_{k=1}^{K} \rho_{ik}\right) - \eta_i \ln(\eta_i) - (1-\eta_i)\ln(1-\eta_i) \right\} \\
&+ \sum_{k=1}^{K}\sum_{\ell=1}^{L} \left\{ \ln\left(\frac{\Gamma(\phi_{1l}+\phi_{2l})}{\Gamma(\phi_{1l})\Gamma(\phi_{2l})}\right) - \ln\left(\frac{\Gamma(\sigma_{1kl}+\sigma_{2kl})}{\Gamma(\sigma_{1kl})\Gamma(\sigma_{2kl})}\right) + \ln\left(\frac{\Gamma(\tau_{1l}+\tau_{2l})}{\Gamma(\tau_{1l})\Gamma(\tau_{2l})}\right) \right. \\
&- \ln\left(\frac{\Gamma(\epsilon_{1kl}+\epsilon_{2kl})}{\Gamma(\epsilon_{1kl})\Gamma(\epsilon_{2kl})}\right) + (\phi_{1l}-\sigma_{1kl})(\psi(\sigma_{1kl}) - \psi(\sigma_{1kl}+\sigma_{2kl})) \\
&+ (\phi_{2l}-\sigma_{2kl})(\psi(\sigma_{2kl}) - \psi(\sigma_{1kl}+\sigma_{2kl})) + (\tau_{1l}-\epsilon_{1kl})(\psi(\epsilon_{1kl}) - \psi(\epsilon_{1kl}+\epsilon_{2kl})) \\
&\left. + (\tau_{2l}-\epsilon_{2kl})(\psi(\epsilon_{2kl}) - \psi(\epsilon_{1kl}+\epsilon_{2kl})) \right\} + \sum_{k=1}^{K-1} \left\{ (\psi(s_1) - \ln(s_2)) + \ln\left(\frac{\Gamma(\kappa_{1k})\Gamma(\kappa_{2k})}{\Gamma(\kappa_{1k}+\kappa_{2k})}\right) \right. \\
&\left. + (\frac{s_1}{s_2} - \kappa_{2k})(\psi(\kappa_{2k}) - \psi(\kappa_{1k}+\kappa_{2k})) - (\kappa_{1k}-1)(\psi(\kappa_{1k}) - \psi(\kappa_{1k}+\kappa_{2k})) \right\} \\
&+ \sum_{k} \left\{ \frac{D}{2}\left[\ln\left(\frac{v}{v_k}\right) - \left(\frac{v}{v_k}\right) + 1\right] - \frac{p}{2}\ln\left(\frac{|\boldsymbol{\Psi}|}{|\boldsymbol{\Psi}_k|}\right) - \frac{p_k}{2}\left[(\boldsymbol{\lambda}_k - \boldsymbol{\lambda}) v \boldsymbol{\Psi}_k^{-1} (\boldsymbol{\lambda}_k - \boldsymbol{\lambda})^\top \right.\right. \\
&\left. + tr(\boldsymbol{\Psi}\boldsymbol{\Psi}_k^{-1}) - D\right] + \ln\left(\frac{\Gamma_D\left(\frac{p_k}{2}\right)}{\Gamma_D\left(\frac{p}{2}\right)}\right) + \frac{1}{2}\left[\ln\frac{|\widehat{\boldsymbol{\Sigma}}_k|}{|\boldsymbol{C}|} - \mathrm{tr}(\boldsymbol{C}^{-1}(\widehat{\boldsymbol{\Sigma}}_k + \widehat{\boldsymbol{\mu}}_k\widehat{\boldsymbol{\mu}}_k^\top))\right] + \frac{D+1}{2} \left.\right\} \\
&+ (\chi_1 - s_1)(\psi(s_1) - \ln(s_2)) + (s_2 - \chi_2)\frac{s_1}{s_2} + \ln\left(\frac{\Gamma(s_1)}{\Gamma(\chi_1)}\right) - s_1\ln(s_2) + \chi_1\ln(\chi_2)
\end{aligned}
\tag{D.2}
$$

References

H. Alagarai Sampath, R. Rajeshuni, and B. Indurkhya. Cognitively inspired task design to improve user performance on crowdsourcing platforms. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3665–3674. ACM, 2014.

M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H.R. Motahari-Nezhad, E. Bertino, and S. Dustdar. Quality control in crowdsourcing systems: Issues and directions. *Internet Computing, IEEE*, 17(2):76–81, March 2013.

O. Alonso. Implementing crowdsourcing-based relevance experimentation: an industrial perspective. *Information retrieval*, 16(2):101–120, 2013.

Amazon. Amazon mechanical turk. `http://www.mturk.com`, 2005.

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

C. E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

Y. Bachrach, T. Graepel, T. Minka, and J. Guiver. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *arXiv preprint arXiv:1206.6386*, 2012.

D. W. Barowy, C. Curtsinger, E. D. Berger, and A. McGregor. Automan: A platform for integrating human-based and digital computation. *ACM SIGPLAN Notices*, 47(10):639–654, 2012.

M. S. Bernstein, D. R. Karger, R. C. Miller, and J. Brandt. Analytic methods for optimizing realtime crowdsourcing. *arXiv preprint arXiv:1204.2995*, 2012.

Y. Bi and D. R. Jeske. The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise. *Journal of Multivariate Analysis*, 101(7):1622–1637, 2010.

L. Biewald and C. Van Pelt. Crowdflower. `http://www.crowdflower.com`, 2007.

D. Blackwell. Discreteness of ferguson selections. *The Annals of Statistics*, 1(2): pp. 356–358, 1973.

D. Blackwell and J. B. Macqueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. In *Bayesian Analysis*, volume 1, pages 121–144, 2006.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

I. Boutsis and V. Kalogeraki. Crowdsourcing under real-time constraints. In *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, pages 753–764. IEEE, 2013.

J. Bragg and D. S. Weld. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI conference on human computation and crowdsourcing*, 2013.

M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality data? *Perspectives on Psychological Science*, 2011.

B. Carpenter. Multilevel bayesian models of categorical data annotation. *Unpublished manuscript*, 2008.

J. L. Carroll, R. Haertel, P. Mcclanahan, and E. Ringger. Modeling the annotation process for ancient corpus creation. In *Chatressar 2007, Proceedings of the International Conference of Electronic Corpora of Ancient Languages (ECAL), Prague, Czech Republic*, 2007.

A. B. Chan. Multivariate generalized gaussian process models. *arXiv preprint arXiv:1311.0360*, 2013.

D. Che, M. Safran, and Z. Peng. From big data to big data mining: challenges, issues, and opportunities. In *Database Systems for Advanced Applications*, pages 1–15. Springer, 2013.

M. Condercet. Essay on the Application of Analysis to the Probability of Majority Decisions, 1785.

P. Dai, C. H. Lin, and D. S. Weld. Pomdp-based control of workflows for crowdsourcing. *Artificial Intelligence*, 202:52–85, 2013.

D. R. Davies, G. Matthews, R. B. Stammers, and S. J. Westerman. *Human performance: Cognition, stress and individual differences.* Psychology Press, 2013.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society*, 28(1):pp. 20–28, 1979.

L. De Alfaro, A. Kulshreshtha, I. Pye, and B. T. Adler. Reputation systems for open collaboration. *Communications of the ACM*, 54(8):81–87, 2011.

B. de Finetti. On the condition of partial exchangeability. *Studies in inductive logic and probability*, 2:193–205, 1980.

N. Della Penna and M. D. Reid. Crowd & prejudice: An impossibility theorem for crowd labelling without a gold standard. *Proceedings of Collective Intelligence 2012*, abs/1204.3511, 2012.

G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478. ACM, 2012.

D. E. Difallah, M. Catasta, G. Demartini, and P. Cudré-Mauroux. Scaling-up the crowd: Micro-task pricing schemes for worker retention and latency improvement. In *Second AAAI Conference on Human Computation and Crowdsourcing difallah-scaleup. pdf Google Scholar BibTex*, 2014.

D. DiPalantino and M. Vojnovic. Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 119–128. ACM, 2009.

J-X. Dong. Statistical results of human performance on usps database, October 2001.

P. Donmez, J. G. Carbonell, and J. G. Schneider. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *SDM*, volume 2, page 1. SIAM, 2010.

S. Doumar and D. Feirstein. Liveops. `http://www.liveops.com`, 2000.

F. Eggenberger and G. Pólya. Über die statistik verketteter vorgänge. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 3(4):279–289, 1923.

C. Eickhoff and A. P. de Vries. Increasing cheat robustness of crowdsourcing tasks. *Information retrieval*, 16(2):121–137, 2013.

Elance-oDesk. Elance-odesk. `http://www.elance-odesk.com`, 2014.

M. D. Escobar. Estimating normal means with a dirichlet process prior. *Journal of the American Statistical Association*, 89(425):pp. 268–277, 1994.

Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):pp. 577–588, 1995.

E. Estellés-Arolas and F. González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200, 2012.

W. Fan and N. Bouguila. Infinite dirichlet mixture models learning via expectation propagation. *Advances in data Analysis and Classification*, 7(4):465–489, 2013.

S. Favaro and Y. W. Teh. Mcmc for normalized random measure mixture models. *Preprint*, 2012.

P. Fearnhead. Particle filters for mixture models with an unknown number of components. *Statistics and Computing*, 14(1):11–21, 2004.

P. Felt, R. Haertel, E. K. Ringger, and K. D. Seppi. Momresp: A bayesian model for multi-annotator document labeling. In *9th edition of the Language Resources and Evaluation Conference*. European Language Resources Association (ELRA), 2014.

T. S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, 1973.

C. Fraly and A. E. Raftery. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588, 1998.

B. Frei. Paid crowdsourcing: Current state & progress toward mainstream business use. *Smartsheet White Paper*, 2009.

P. G. Moreno, Y. W. Teh, F. Perez-Cruz, and A. Artés-Rodríguez. Bayesian nonparametric crowdsourcing. *The Journal of Machine Learning Research*, -(-): –, 2014. To appear.

P. G. Moreno, A. Artés-Rodríguez, and F. Perez-Cruz. A nonparametric bayesian model for the multiple annotators problem. *IEEE Transactions on Neural Networks and Learning Systems*, -(-):–, 2015. Submitted.

J. Gantz and D. Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. 2012.

T. George and S. Merugu. A scalable collaborative filtering framework based on co-clustering. In *Data Mining, Fifth IEEE International Conference on*, pages 4 pp.–, Nov 2005. doi: 10.1109/ICDM.2005.14.

S. J. Gershman and D. M. Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

Z. Ghahramani and M. J. Beal. Propagation algorithms for variational bayesian learning. *Advances in neural information processing systems*, pages 507–513, 2001.

Z. Ghahramani and H-C. Kim. Bayesian classifier combination. *Technical Report*, 2003.

Z. Ghahramani, M. I. Jordan, and R. P. Adams. Tree-structured stick breaking for hierarchical data. In *Advances in neural information processing systems*, pages 19–27, 2010.

H. W. Gould. Stirling number representation problems. *Proceedings of the American Mathematical Society*, 11(3):447–451, 1960.

P. J. Green and S. Richardson. Modelling heterogeneity with and without the dirichlet process. *Scandinavian journal of statistics*, 28(2):355–375, 2001.

T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. In *Advances in neural information processing systems*. MIT Press, 2005.

P. Groot, A. Birlutiu, and T. Heskes. Learning from multiple annotators with gaussian processes. In *Proc. of the 21st International Conference on Artificial Neural Networks - Volume Part II*, pages 159–164, 2011.

A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.

L. Hannah, D. Blei, and W. Powell. Dirichlet process mixtures of generalized linear models. In *Artificial Intelligence and Statistics*, 2010.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

D. Have and D. Elley. Ponoko. `http://www.ponoko.com`, 2007.

K. Heung-Nam, J. Ae-Ttie, H. Inay, and J Geun-Sik. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, 9(1):73 – 83, 2010. ISSN 1567-4223. Special Issue: Social Networks and Web 2.0.

E. Hewitt and L. J. Savage. Symmetric measures on cartesian products. *Transactions of the American Mathematical Society*, pages 470–501, 1955.

N.L. Hjort, C. Holmes, P. Müller, and S.G. Walker. *Bayesian nonparametrics*. Cambridge University Press, 1st edition, 2010.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 209–218. ACM, 2010.

T. Hoßfeld, M. Hirth, J. Redi, F. Mazza, P. Korshunov, B. Naderi, M. Seufert, B. Gardlo, S. Egger, and C. Keimel. Best Practices and Recommendations for Crowdsourced QoE - Lessons learned from the Qualinet Task Force "Crowdsourcing", October 2014.

D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. H. Hovy. Learning whom to trust with mace. In *HLT-NAACL*, pages 1120–1130, 2013.

J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

C. Hurley, S. Chen, and J. Karim. Youtube. `http://www.youtube.com`, 2005.

P. G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):16–21, 2010.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):pp. 161–173, 2001.

S. Jain and R. M. Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1), 2004.

S. Jain and R. M Neal. Splitting and merging components of a nonconjugate dirichlet process mixture model. *Bayesian Analysis*, 2(3):445–472, 2007.

E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, April 2003. ISBN 0521592712.

M. Jordan and T. Jaakkola. A Variational Approach to Bayesian Logistic Regression Models and Their Extensions. In *Workshop on Artificial Intelligence and Statistics*, 1996.

M. I Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

H. Kajino and H. Kashima. Convex formulations of learning from crowds. *Transactions of the Japanese Society for Artificial Intelligence*, 27:133–142, 2012.

H. Kajino, Y. Tsubo, and H. Kashima. Clustering crowds. In *27th AAAI Conference on Artificial Intelligence*, pages 1120–1127, 2013.

M. Kalli, J. E. Griffin, and S. G. Walker. Slice sampling mixture models. *Statistics and Computing*, 21(1):93–105, January 2011.

E. Kamar and E. Horvitz. Planning for crowdsourcing hierarchical tasks. In *Proceedings of international conference on autonomous agents and multiagent systems*, 2015.

D. R Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk. In *AMCIS*, volume 11, pages 1–11, 2011.

G. Kazai. An exploration of the influence that task parameters have on the performance of crowds. *Proceedings of the CrowdConf*, 2010, 2010.

C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

S. Khanna, A. Ratan, J. Davis, and W. Thies. Evaluating and improving the usability of mechanical turk for low-income workers in india. In *Proceedings of the first ACM symposium on computing for development*, page 12. ACM, 2010.

F. Khan Khattak and A. Salleb-Aouissi. Quality control of crowd labeling through expert evaluation. In *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, 2011.

H-C. Kim and Z. Ghahramani. Bayesian classifier combination. *Journal of Machine Learning Research*, 22:619–627, 2012.

A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 43–52. ACM, 2011.

A. Kittur, J.V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton. The future of crowd work. In *Proceedings of the 2013*

conference on Computer supported cooperative work, pages 1301–1318. ACM, 2013.

Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

Andreĭ Nikolaevich Kolmogorov. *Foundations of the Theory of Probability.* Chelsea Publishing Co., 1950.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263.

A. Kulkarni, M. Can, and B. Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1003–1012. ACM, 2012.

K. Kurihara, M. Welling, and N. Vlassis. Accelerated variational dirichlet process mixtures. In *Neural Information Processing Systems*, 2006.

K. Kurihara, M. Welling, and Y. W. Teh. Collapsed Variational Dirichlet Process Mixture Models. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2007.

P. A. Lachenbruch. Discriminant analysis when the initial samples are misclassified. *Technometrics*, 8(4):657–662, 1966.

P. A. Lachenbruch. Note on initial misclassification effects on the quadratic discriminant function. *Technometrics*, 21(1):129–132, 1979.

C. P. Lam and D. G. Stork. Toward optimal labeling strategy under multiple unreliable labelers. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pages 42–47, 2005.

W. S. Lasecki and J. P. Bigham. Interactive crowds: real-time crowdsourcing and crowd agents. In *Handbook of Human Computation*, pages 509–521. Springer, 2013.

J. Lee, W. Kladwang, D. Lee, D. Cantu, M. Azizyan, H. Kim, A. Limpaecher, S. Yoon, A. Treuille, and R. Das. Eterna. `http://eterna.cmu.edu/web/`, 2014.

K. Lee, J. Caverlee, and S. Webb. The social honeypot project: protecting online communities from spammers. In *Proceedings of the 19th international conference on World wide web*, pages 1139–1140. ACM, 2010.

S. Lee, S. Yildirim, A. Kazemzadeh, and S. Narayanan. An articulatory study of emotional speech production. In *INTERSPEECH*, pages 497–500. ISCA, 2005.

C. W. Leung, S. C. Chan, and F. L. Chung. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge Based Systems*, 21(7):515–529, October 2008. ISSN 0950-7051.

J. C. R. Licklider. Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, 14(1):4–11, 1960.

G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 57–66. ACM, 2010.

J. S. Liu. Nonparametric hierarchical bayes via sequential imputations. *The Annals of Statistics*, pages 911–930, 1996.

J. S. Liu, W. H. Wong, and A. Kong. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81(1):27–40, March 1994.

Q. Liu, J. Peng, and A. T. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 692–700, 2012.

S. Loh, F. Lorenzi, R. Granada, D. Lichtnow, L. K. Wives, and J. P. M. de Oliveira. Identifying similar users by their scientific publications to reduce cold start in recommender systems. In *WEBIST*, pages 593–600. INSTICC Press, 2009.

X. Luo, Y. Xia, and Q. Zhu. Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, 27:271–280, March 2012. ISSN 0950-7051.

S. N. MacEachern. Estimating normal means with a conjugate style dirichlet process prior. *Communications in Statistics-Simulation and Computation*, 23 (3):727–741, 1994.

S. N. MacEachern and P. Müller. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):pp. 223–238, 1998.

S. N. MacEachern, M. Clyde, and J. S. Liu. Sequential importance sampling for nonparametric bayes models: The next generation. *Canadian Journal of Statistics*, 27(2):251–267, 1999.

W. Mason and D. J. Watts. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108, 2010.

R. A. McDonald, D. J. Hand, and I. A. Eckley. An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *Multiple Classifier Systems*, pages 35–44. Springer, 2003.

N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

J. W. Miller and M. T. Harrison. A simple example of Dirichlet process mixture inconsistency for the number of components. In *Neural Information Processing Systems*, 2013.

T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

H. Moravec. When will computer hardware match the human brain. *Journal of evolution and technology*, 1(1):10, 1998.

P. Muliere and L. Tardella. Approximating distributions of random functionals of ferguson-dirichlet priors. *Canadian Journal of Statistics*, 26(2):283–297, 1998.

R. T. Nakatsu, E. B. Grossman, and C. L. Iacovou. A taxonomy of crowdsourcing based on task complexity. *Journal of Information Science*, page 0165551514550140, 2014.

A. Nazabal, P. G. Moreno, A. Artes-Rodriguez, and Z. Ghahramani. Human activity recognition by combining a small number of classifiers. *Journal of Biomedical and Health Informatics*, -(-):–, 2015. To appear.

R. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

R. M. Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

D. F. Nettleton, A. Orriols-Puig, and A. Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review*, 33(4):275–306, 2010.

P. Orbanz and Y.W. Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*. Springer, 2010.

J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical dirichlet processes. *arXiv preprint arXiv:1210.6738*, 2012.

O. Papaspiliopoulos. A note on posterior sampling from dirichlet mixture models. Working paper, University of Warwick. Centre for Research in Statistical Methodology, 2008.

O. Papaspiliopoulos and G. O. Roberts. Retrospective markov chain monte carlo methods for dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, 2008.

S. Patterson and Y. W. Teh. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, pages 71–110, 1894.

J. Pitman. Combinatorial stochastic processes. Technical Report 621, Dept. Statistics, U.C. Berkeley, 2002. Lecture notes for St. Flour course.

J. Pitman and M. Yor. The Two-Parameter Poisson-Dirichlet Distribution Derived from a Stable Subordinator. *The Annals of Probability*, 25(2), 1997.

Y. Raimond, T. Ferne, M. Smethurst, and G. Adams. The bbc world service archive prototype. *Web Semantics: Science, Services and Agents on the World Wide Web*, 27:2–9, 2014.

C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

V. C. Raykar and S. Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, pages 491–518, 2012.

V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 99:1297–1322, 2010.

A. Renyi. *Probability theory*. North-Holland Publishing. Co, 1970.

David Rindskopf and Wallace Rindskopf. The value of latent class analysis in medical diagnosis. *Statistics in Medicine*, 5(1):21–27, 1986.

F. Rodrigues, F. Pereira, and B. Ribeiro. Learning from multiple annotators: Distinguishing good from random labelers. *Pattern Recognition Letters*, 34(12): 1428–1436, 2013.

F. Rodrigues, F. Pereira, and B. Ribeiro. Gaussian process classification and active learning with multiple annotators. In *Proceedings of The 31st International Conference on Machine Learning*, pages 433–441. Journal of Machine Learning Research, 2014.

A. Rodriguez and K. Ghosh. Nested partition models. *Jack Baskin School of Engineering*, 2009.

J. Rogstadius, V. Kostakos, A. Kittur, B. Smus, J. Laredo, and M. Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *ICWSM*, 2011.

C. Rozsenich, G. Kresin, A. Conic, D. Marz, and I. Maione. Clickworker. `http://www.clickworker.com`, 2005.

P. Ruvolo, J. Whitehill, and J. R. Movellan. Exploiting commonality and interaction effects in crowdsourcing tasks using latent factor models. In *Neural Information Processing Systems. Workshop on Crowdsourcing: Theory, Algorithms and Applications*, 2013.

C. Ryll-Nardzewski. On stationary sequences of random variables and the de finetti's equivalence. *Colloquium Mathematicae*, 4(2):149–156, 1957.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4: 639–650, 1994.

B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

B. Shahbaba and R. Neal. Nonlinear models using dirichlet process mixtures. *Journal of Machine Learning Research*, 10:1829–1850, 2009.

L. Shang and A. B. Chan. On approximate inference for generalized gaussian process models. *arXiv preprint arXiv:1311.6371*, 2013.

Y. Shi, M. Larson, and A. Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, 47(1):3:1–3:45, May 2014.

E. Simpson, S. J. Roberts, A. Smith, and C. Lintott. Bayesian combination of multiple, imperfect classifiers. In *Neural Information Processing Systems*, pages 1–8, Oxford, 2011. University of Oxford.

E. Simpson, S. Roberts, I. Psorakis, and A. Smith. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*, volume 474 of *Studies in Computational Intelligence*, pages 1–35. Springer, 2013.

E. Simpson, M. Venanzi, S. Reece, P. Kohli, J. Guiver, S. Roberts, and N. Jennings. Language understanding in the wild: Combining crowdsourcing and machine learning. In *24th International World Wide Web Conference*, 2015.

R. Simpson, K. R. Page, and D. De Roure. Zooniverse: observing the world's largest citizen science platform. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 1049–1054. International World Wide Web Conferences Steering Committee, 2014.

Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1157–1166. International World Wide Web Conferences Steering Committee, 2013.

P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. *Advances in neural information processing systems*, pages 1085–1092, 1995.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language

tasks. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

K.R. Steingart, M. Henry, V. Ng, P.C. Hopewell, A. Ramsay, J. Cunningham, R. Urbanczik, M. Perkins, M. Abdel Aziz, and M. Pai. Fluorescence versus conventional sputum smear microscopy for tuberculosis: a systematic review. *The Lancet Infectious Diseases*, 6(9):570–581, 2006.

M. Stone. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

X. Su and T. M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence.*, 2009:1–19, January 2009.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101, 2003.

Y. Tian and J. Zhu. Learning from crowds in the presence of schools of thought. In *Knowledge Discovery and Data Mining*, pages 226–234, 2012.

L. Tran-Thanh, T. D. Huynh, A. Rosenfeld, and N. R. Ramchurn, S. D .and Jennings. Budgetfix: budget limited crowdsourcing for interdependent task allocation with quality guarantees. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 477–484. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

L. Tran-Thanh, T. D. Huynh, A. Rosenfeld, S. D. Ramchurn, and N. R. Jennings. Crowdsourcing complex workflows under budget constraints. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1298–1304. AAAI Press, 2015.

A. M. Turing. Computing machinery and intelligence. *Mind*, pages 433–460, 1950.

Yener Ülker, Bilge Günsel, and Ali T Cemgil. Sequential monte carlo samplers for

dirichlet process mixtures. In *International Conference on Artificial Intelligence and Statistics*, pages 876–883, 2010.

Yener Ulker, Bilge Gunsel, and Ali Taylan Cemgil. Annealed smc samplers for nonparametric bayesian mixture models. *Signal Processing Letters, IEEE*, 18 (1):3–6, 2011.

I. Ulusoy and C. M. Bishop. Generative versus discriminative methods for object recognition. *Computer Vision and Pattern Recognition, 2005*, 2:258–265, 2005.

M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164. International World Wide Web Conferences Steering Committee, 2014.

D. Ververidis and C. Kotropoulos. Emotional speech recognition: Resources, features, and methods. *Speech Communication*, 48(9):1162–1181, September 2006.

L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

L. Von Ahn, B. Maurer, Colin McMillen, D. Abraham, and M. Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321 (5895):1465–1468, 2008.

L. Von Ahn, S. Hacker, A. Navas, V. Cheung, M. Uekermann, B. Meeder, H. Villafuerte, and J Fuentes. Duolingos. `https://www.duolingo.com`, 2012.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2): 1–305, 2008.

S. G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation*, 2007.

C. Wang and D. M. Blei. Truncation-free online variational inference for bayesian nonparametric models. In *Advances in neural information processing systems*, pages 413–421, 2012.

C. Wang, X. Liao, L. Carin, and D. B. Dunson. Classification with incomplete data using dirichlet process priors. *Journal of Machine Learning Research*, 9999: 3269–3311, December 2010.

J. Wang, P. G. Ipeirotis, and F. Provost. Managing crowdsourcing workers. In *The 2011 Winter Conference on Business Intelligence*, pages 10–12, 2011.

J. Wang, P. G. Ipeirotis, and F. Provost. Quality-based pricing for crowdsourced workers. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1157–1166. International World Wide Web Conferences Steering Committee, 2013.

F. L. Wauthier and M. I. Jordan. Bayesian bias mitigation for crowdsourcing. In *Neural Information Processing Systems*, pages 1800–1808, 2011.

P. Welinder, S. Branson, S. Belongie, and P. Perona. The Multidimensional Wisdom of Crowds. In *Neural Information Processing Systems*, pages 2424–2432, 2010.

L. T. Weng, Yue X., Yuefeng L., and R. Nayak. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on*, volume 2, pages 113–120, Nov 2008. doi: 10.1109/ICTAI.2008.97.

M. West, P. Muller, and M. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. In P. Freeman and A. Smith, editors, *Aspects of Uncertainty*, pages 363–386. John Wiley, 1994.

T. White. *Hadoop: The definitive guide.* " O'Reilly Media, Inc.", 2012.

J. Whitehill, T. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown

expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

Wikimedia Foundation. Wikipedia. `http://www.wikipedia.org`, 2001.

O. Wu, W. Hu, and J. Gao. Learning to rank under multiple annotators. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 2 of *IJCAI'11*, pages 1571–1576. AAAI Press, 2011.

Yahoo Inc. Yahoo! answers. `https://answers.yahoo.com`, 2005.

Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo, L. Bogoni, L. Moy, and J. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. *International conference on artificial intelligence and statistics*, pages 932–939, 2010.

Y. Yan, G. M. Fung, R. Rosales, and J. G. Dy. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1161–1168, 2011.

Y. Yan, R. Rosales, G. Fung, and J. Dy. Modeling multiple annotator expertise in the semi-supervised learning scenario. *arXiv preprint arXiv:1203.3529*, 2012.

M.A. Young, R. Abrams, M.A. Taylor, and H.Y. Meltzer. Establishing diagnostic criteria for mania. *The Journal of Nervous and Mental Disease*, 171(11):676–682, 1983.

M-C. Yuen, I. King, and K-S. Leung. A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 766–773. IEEE, 2011.

S. E. Yuksel, J. N. Wilson, and P. D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012.

RuLong Z. and SongJie G. Analyzing of collaborative filtering using clustering technology. In *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*, volume 4, pages 57–59, Aug 2009. doi: 10.1109/CCCM.2009.5267822.

Sheng Z., Weihong W., J. Ford, F. Makedon, and J. Pearlman. Using singular value decomposition approximation for collaborative filtering. In *E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on*, pages 257–264, July 2005. doi: 10.1109/ICECT.2005.102.

M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, volume 10, page 10, 2010.

P. Zhang and Z. Obradovic. Learning from inconsistent and unreliable annotators by a gaussian mixture model and bayesian information criterion. In *Proc. of the ECML/PKDD 2011*, 2011.

Y. Zhang and M. van der Schaar. Reputation-based incentive protocols in crowdsourcing applications. In *INFOCOM, 2012 Proceedings IEEE*, pages 2140–2148. IEEE, 2012.

Y. Zhao and Q. Zhu. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers*, 16(3):417–434, 2014.

D. Zhou, S. Basu, Y. Mao, and J. C. Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203, 2012.

X Zhu and X Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004.

O. Zobay. Mean field inference for the dirichlet process mixture model. *Electronic Journal of Statistics*, 3:507–545, 2009.